



Technická univerzita v Liberci

FAKULTA MECHATRONIKY A MEZIOBOROVÝCH INŽENÝRSKÝCH STUDIÍ

DIPLOMOVÁ PRÁCE

Automatizované měřicí pracoviště pro statické a dynamické vážení

Automated workstation of measuring for static and dynamic weighting

Vedoucí práce: Ing. Lenka Kretschmerová

Konzultant: Ing. Jakub Wiener Ph.D

Liberec 2003

Jan Horák

ANOTACE

Autor:	Jan Horák
Diplomová práce:	Automatizované měřicí pracoviště pro statické a dynamické vážení
Vedoucí diplomové práce:	Ing. Lenka Kretschmerová
Rok publikace:	2003

Resumé:

Presentovaná diplomová práce se zabývá problémem automatického sběru dat z digitálních vah firmy KERN. Úlohou této práce bylo vytvořit program v systému Microsoft Windows, který bude schopen sběru dat ze sériového portu, zobrazování dat v grafech a jejich ukládání na disk pro potřebu pozdějšího zpracování. Výsledkem této práce je na CD nosiči přiložená aplikace, implementující tyto funkce, která bude jistě přínosem pro další výzkum vztlínání kapaliny textiliemi na Katedře zušlechťování materiálů Textilní fakulty Technické univerzity Liberec.

ANOTATION

Author:	Jan Horák
Diploma Thesis:	Automated workstation of measuring for static and dynamic weighting
Supervisor:	Ing. Lenka Kretschmerová
Year of publication:	2003

Resume:

Presented Diploma Thesis deals with problem of automated data acquisition from digital scales by firm KERN. The task of this diploma was to create a program in system Microsoft Windows, which would be able to collect data from the serial port, showing it in charts and saving to hard drive for further treatment.

The result of this work is an application, enclosed on CD, which implements functions that are enumerated above. This Application will be certainly asset for the advancement of the research of capillary action on the Department of Textile Finishing of Faculty of Textile Engineering.

Poděkování

Také touto cestou bych chtěl poděkovat vedoucí diplomové práce ing. Lence Kretschmerové a konzultantu ing. Jakubu Wienerovi Ph.D za jejich pomoc, návrhy a konstruktivní rady při řešení této diplomové práce.

OBSAH

1. ÚVOD	9
2. ANALÝZA PROBLÉMU	10
2.1. DIGITÁLNÍ VÁHY <i>KERN 770 GJ</i>	10
2.1.1. TECHNICKÉ SPECIFIKACE VAH:	11
2.1.2. OVLÁDÁNÍ MENU	11
2.1.3. PARAMETRY SÉRIOVÉHO PŘENOSU	12
2.1.4. FORMÁT VÝSTUPNÍCH DAT	12
2.1.5. FORMÁT VSTUPNÍCH DAT	13
2.1.6. FORMÁT PŘÍKAZŮ	13
2.2. KOMUNIKACE VAH S POČÍTAČEM	13
2.2.1. KOMUNIKACE S POČÍTAČEM	14
2.2.2. NASTAVENÍ KOMUNIKACE PROGRAMU S VAHAMI	14
2.2.3. STATICKÉ VÁŽENÍ	14
2.2.4. DYNAMICKÉ VÁŽENÍ	14
2.2.5. MATEMATICKÁ ANALÝZA	15
3. IMPLEMENTACE PROGRAMU	16
3.1. SÉRIOVÉ ROZHRANÍ RS-232-C	16
3.1.1. ELEKTRICKÉ PARAMETRY	16
3.1.2. SIGNÁLY ROZHRANÍ A PARAMETRY PŘENOSU	16
3.2.3. KONKRÉTNÍ ZAPOJENÍ ROZHRANÍ U VAH <i>KERN 770</i>	17
3.2.4. PROGRAMOVÁNÍ RS-232-C POMOCÍ WINDOWS API	18
3.2.5. PROCEDURA PŘIPOJENÍ SÉRIOVÉHO PORTU	20
3.2.6. ODPOJENÍ SÉRIOVÉHO PORTU	22
3.2.7. ČTENÍ DAT ZE SÉRIOVÉHO KANÁLU	22
3.2.8. KONVERZE NAČTENÝCH DAT DO FORMÁTU S PLOVOUCÍ DESETINNOU ČÁRKOU	24
3.2. UŽIVATELSKÉ ROZHRANÍ PROGRAMU	25
3.2.1. SPUŠTĚNÍ APLIKACE	25
3.2.2. ČÁST PROGRAMU <i>STATICKÉ MĚŘENÍ</i>	25
3.2.3. ČÁST PROGRAMU <i>DYNAMICKÉ MĚŘENÍ</i>	26
3.2.4. MATEMATICKÁ ANALÝZA	28
3.2.5. NASTAVENÍ PROGRAMU A TISK	33

4. PROCES VZLÍNÁNÍ KAPALINY.....	34
4.1. TEORETICKÝ ÚVOD	34
4.2. EXPERIMENTÁLNÍ USPOŘÁDÁNÍ.....	38
4.3. GRAFY.....	40
4.3.1. NAMĚŘENÁ DATA PROLOŽENÁ S ODMOCNINOVOU APROXIMACÍ.	40
4.3.2. NAMĚŘENÁ DATA PROLOŽENÁ BEZ APROXIMACE.	41
5. ZÁVĚR.....	42
PŘÍLOHA A. UŽIVATELSKÁ PŘÍRUČKA	43
A.1. INSTALACE A NASTAVENÍ PROGRAMU.....	43
A.1.1. NASTAVENÍ PORTU RS-232-C VAH.....	43
A.1.2. NASTAVENÍ PARAMETRŮ SÉRIOVÉ LINKY PROGRAMU.....	44
A.2. STATICKÉ MĚŘENÍ.....	45
A.3. DYNAMICKÉ MĚŘENÍ	47
A.4. MATEMATICKÁ ANALÝZA.....	49
A.5. TISK A NASTAVENÍ TISKÁRNY	51
A.6. HARDWAROVÉ NÁROKY	52
A.7. ELEKTRONICKÝ NOSIČ	52
PŘÍLOHA B. DŮLEŽITÉ ČÁSTI KÓDU	53
B.1. PŘIPOJENÍ SÉRIOVÉHO KANÁLU.....	53
B.2. SBĚR DAT ZE SÉRIOVÉHO KANÁLU	54
B.3. KONVERZE DAT DO FORMÁTU S PLOVOUCÍ DESETINNOU ČÁRKOU	54
B.4. UDÁLOSTNÍ KÓD DIALOGOVÉHO TLAČÍTKA ZAZNAMENAT	55
B.5. KÓD METODY NEJMENŠÍCH ČTVERCŮ.....	55
B.6. KÓD METODY DĚLENÍ INTERVALU.....	56
PŘÍLOHA C. ELEKTRONICKÝ NOSIČ	57
POUŽITÁ LITERATURA	58

1. Úvod

Díky obrovskému technologickému pokroku v oblasti číslicové elektroniky a měřicích systémů se v poslední době klade velký důraz na rozvoj počítačem řízených měřicích systémů, do procesu měření stále více proniká automatizace. Automatizované systémy sběru dat přinášejí do oboru měření nové výhody, počínaje zjednodušením obsluhy přístrojů a odstraněním náhodných chyb, vzniklých špatným odečítáním hodnot, po možnost samostatného běhu přístrojů bez nutnosti lidského zásahu.

Rozvoj osobních počítačů klade stále větší nároky na měřicí přístroje, vznikají přístroje s integrovanými sběrnicemi pro připojení k vnějším systémům a sítím. U levnějších přístrojů se nejčastěji objevuje sériový kanál RS-232-C, jehož architektura a vlastnosti umožňují skutečně levné a jednoduché propojení mezi přístroji, většinou měřicího přístroje s osobním počítačem.

Nejinak je tomu u laboratorních digitálních vah KERN 770, které umožňují připojení přes sériový port a sběr dat, které reprezentují hmotnost měřeného přípravku. Cílem mé práce je návrh, algoritmizace a implementace automatizované aplikace v prostředí Microsoft Windows, která bude schopna automaticky získávat data z digitálních vah, ukládat k pozdějšímu zpracování a provádět matematickou analýzu jevu kapilárního vztlínání kapaliny v textiliích.

Práce se skládá ze čtyř částí. První část obsahuje analýzu problému spojení digitálních vah s osobním počítačem, druhá obsahuje detailní popis částí programu a použitých algoritmů s důrazem na problémy systému Microsoft Windows. Třetí část popisuje experiment vztlínání kapaliny vláknem a konečně poslední, čtvrtá část, je uživatelská příručka pro obsluhu vlastní aplikace.

2. Analýza problému

V následující části si představíme digitální váhy od německé firmy *Kern Und Sohn*, typ *Kern 770 GJ*, popíšeme si princip, který používají pro připojení k vnějším zařízením, jako je programovatelný automat, počítač nebo tiskárna. Dále popíšeme nastin funkcí, které by měl náš program mít a nadefinujeme si nutné softwarové nástroje, které bude nutno použít pro vývin obslužné aplikace vah.

2.1. Digitální váhy *KERN 770 GJ*



Obrázek.2.1. Digitální váhy *Kern 770 GJ*.
(<http://www.kern-sohn.com/images/waagen/770.jpg>)

KERN 770 GJ jsou velmi přesné digitální váhy určené k provozu v laboratorních podmínkách. Umožňují velké množství funkcí, jako je volba použitých jednotek hmotnosti, automatickou kalibraci a připojení k externím zařízením pomocí *RS – 232-C*. Pro lepší izolaci měřeného vzorku od prostředí jsou vybaveny snímatelným skleněným rámem, pro přístup ke vzorku lze dvě stěny a vršek krytu odsunout.

Vlastní vážení, kalibrace a nastavování jednotek se provádí pomocí šesti tlačítek na předním panelu přístroje, naměřená data a hlášení vah se zobrazují na osmimístném LCD displeji. Váhy jsou napájeny stejnosměrným proudem o napětí 12V. Pro komunikaci s PC nebo pro tisk naměřených hodnot na tiskárně je váha vzadu opatřena standardním devítipinovým sériovým portem.

2.1.1. Technické specifikace vah:

Tabulka 2.1. Technické specifikace vah.

Rozsah	120 g
Rozlišení	0.1 mg
Linearita	0.2 mg
Doba ustálení	3 sec
Přenosová rychlost	150 – 19200 baudů

Nastavování vah a kalibrace se provádí pomocí jednoduchého menu přístupného pomocí tlačítek *CAL*, *PRINT* a *TARE*. Tento postup není příliš uživatelsky přístupný, protože uživatel musí při nastavování hledat příslušné kódy menu v manuálu. S pomocí menu se dají nastavit různé vlastnosti, jako jsou podmínky okolního prostředí (úroveň vibrací), zobrazování dat na displeji, použité jednotky hmotnosti, spouštění aplikačních programů a také nastavení sériové linky a formátu výstupních dat.

Váhy mají dále aplikační programy pro speciální vážení, které usnadňují některé častěji se opakující úkony během vážení. Jsou to:

- celkové vážení,
- vážení v procentech,
- sčítání malých částí,
- průměrování hodnot / vážení živých zvířat.

2.1.2. Ovládání menu

Přístup k nastavovacímu menu vah je umožněn pomocí tlačítek hlavního panelu *CAL* a *PRINT*, kterými se na displeji nastaví kód menu, a tlačítka *TARE*, kterým se nastavení potvrdí. Například pro změnu výstupních jednotek na gramy uvedeme pomocí tlačítek *CAL* a *PRINT* kód menu 1 7 1 a posléze potvrdíme stisknutím tlačítka *TARE*. Kódy jednotlivých menu nastavení jsou uvedeny v manuálu vah.

2.1.3. Parametry sériového přenosu

Ke komunikaci s počítačem jsou váhy *KERN 770 GJ* opatřeny sériovým portem specifikace *RS-232-C* v pětivodičovém zapojení. Toto rozhraní umožňuje softwarový (*Xon/Xoff*) i hardwarový (*CTS/DTR*) handshake, asynchronní full-duplexní komunikaci o nastavitelné přenosové rychlosti 150 - 19200 baudů. Pro komunikaci s vahami jsem pro náš program vybral jako nejvýhodnější princip hardwarového handshaku s řídícími signály mezi *CTS* a *DTR*.

2.1.4. Formát výstupních dat

Pro výstup naměřených hodnot váhy používají dva formáty dat (opět záleží na nastavení menu) a to naměřenou hodnotu s identifikátorem a naměřenou hodnotu bez identifikátoru. Formát s identifikátorem má délku 22 osmibitových znaků, formát bez identifikátoru má délku 16 osmibitových znaků a je zobrazen v tabulce 2.2.

Tabulka 2.2. Formát výstupních dat bez identifikátoru.

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
+															
*	H O D N O T Y									*	U	U	U	CR	LF
-															

* - mezera (ASCII 32)

Význam jednotlivých Bytů:

1 - znaménko

2-10 - naměřená hodnota (včetně desetinné čárky)

12-14 - použité jednotky (g, kg...)

15-16 - ASCII 10 a ASCII 13 - znaky pro konec řádku.

U formátu dat s identifikátorem se před vlastní šestnáctibytový řetězec vkládá ještě šest bytů identifikátoru, které rozlišují výstupní data. Tento identifikátor je nastaven na hodnotu vnitřního čítače vah a jednoznačně určuje pořadí výstupních hodnot. Ostatní byty se tedy posunují o šest míst doprava.

Pro aplikaci, umožňující komunikaci s vahami je výhodnější použít první typ formátu dat, protože aplikace může data spolehlivě rozlišit podle času příchodu na sériový port počítače.

2.1.5. Formát vstupních dat

Pro simulaci stisku tlačítek vah, bohužel ne pro vstup do ovládacího menu vah, přes sériový port se používají podobně, jako je to u tiskáren, escape příkazy, které jsou opět ukončeny znaky ASCII 10 a ASCII 13. Každé tlačítko má svoje číslo escape příkazu.

2.1.6. Formát příkazů

S pomocí řídicích příkazů se simuluje stisk jednotlivých kláves hlavního panelu a tím i ovládání vah. Například posláním příkazu, který simuluje stisk tlačítka *PRINT* váhy na sériový port pošlou hodnotu, která je zobrazena na displeji. Tento způsob ovládání přenosu dat jsem uvažoval jako možný způsob čtení naměřených hodnot počítačem přes sériový port, ale váhy též umožňují stálé posílání dat na sériový port bez přerušení a bez požádání, což je jednodušší a pro náš případ i použitelnější. V tabulce 2.4. jsou uvedeny příkazy pro simulaci nejběžněji používaných kláves [1], [3].

Tabulka 2.4. příkazy pro simulaci stisku jednotlivých kláves.

CAL	ESC	F	1	' _ '	CR	LF
TARE	ESC	T	CR	LF		
PRINT	ESC	P	CR	LF		

2.2. Komunikace vah s počítačem

Pro komunikaci vah s počítačem pomocí standardní sériové linky jsem navrhl program v systému Microsoft Windows. Program jsem vytvořil v systému Inprise Delphi verze 5. Zahrnuje nastavení samotného měření a vlastní měření, které bude moci probíhat i automaticky.

2.2.1. Komunikace s počítačem

Ke komunikaci s počítačem jsem použil sériový port, jehož vlastnosti a rychlost komunikace se nastaví pomocí menu vah a nastavení samotného programu. Pro statické i dynamické vážení bude stačit přenosová rychlost 1200 baudů, která je doporučena od výrobce, jelikož maximální frekvence měření není vyšší než 10 Hz. Rovněž formát znaků a paritu není třeba měnit.

Při zahájení komunikace systém nastaví parametry sériové linky na straně počítače podle přání uživatele tak, aby byl zabezpečen správný přenos dat. Poté bude ze sériového portu číst hmotnost naměřenou vahami, zkonvertuje ji do formátu s plovoucí desetinnou čárkou a bude ji průběžně zobrazovat na obrazovce.

2.2.2. Nastavení komunikace programu s vahami

Na straně programu bude nutno určit správné parametry sériové linky tak, aby korespondovala s nastavením sériové linky na straně vah a portem počítače, který je použitý ke komunikaci.

2.2.3. Statické vážení

Při statickém měření je třeba, aby se naměřená hodnota průběžně zobrazovala v příslušném okně, a proto se bude naměřená hodnota s frekvencí přibližně 5 Hz stále odečítat ze sériového portu. Tato frekvence odečítání by měla být nastavitelná uživatelem. Pro rychlejší a komfortnější zapisování naměřených hodnot bude program opatřen tlačítkem, jehož stisknutím se aktuální hodnota zobrazená v okně přidá do seznamu. Seznam naměřených hodnot si uživatel může opatřovat poznámkami a po skončení měření uložit ve formátu, který je přenositelný do programů Microsoft Excel a Matlab. Bohužel systém vah nedovoluje využít přednastavené programy vážení, o kterých jsem se zmínil v kapitole Technické specifikace vah, jako je sčítání malých částí nebo vážení v procentech.

2.2.4. Dynamické vážení

Dynamické vážení je do programu implementováno z důvodů potřeby měřit dlouhotrvající děje, jako je vysychání látek nebo nasákavost vláken, které budou zmíněny níže. Systém bude schopen automatického sběru dat z vah s nastavitelnou dobou měření a nastavitelnou frekvencí měření. Bude možno měření předčasně ukončit, či jen pozastavit a později znovu pokračovat.

Naměřené hodnoty se budou průběžně zakreslovat do grafu, rozsah jeho os se bude již při probíhající měření moci ručně měnit, čímž bude mít uživatel podrobný přehled o právě probíhající ději.

Data, která budou reprezentovat údaje o hmotnosti naměřená váhami se budou do souboru ukládat spolu s časem jejich příchodu na sériový port počítače, což umožní pozdější rekonstrukci naměřeného průběhu hmotnosti jak v sekci programu nazvané *Matematická analýza*, tak v jiných programech, jako jsou *Microsoft Excel* nebo *Matlab* od firmy *Mathworks*. Minimální čas mezi dvěma uloženými hodnotami je dán frekvencí analogově číslicového převodníku vah, která je 10Hz, takže mezi dvěma sousedními uloženými hodnotami může být minimální časový interval desetina sekundy.

Frekvence ukládání dat do souboru a zobrazování v grafu bude nastavitelná dvěma způsoby:

- Přímým nastavením časového intervalu. Tím bude mít konstantní délku.
- Nastavením změny hmotnosti. V tomto případě bude systém odměřovat hodnoty stále, uloží se ale pouze takové, jejichž velikost se bude od předchozí hodnoty v absolutní hodnotě lišit minimálně o nastavený interval. Tím se zachytí rychlé změny hmotnosti, zatímco u pomalých změn bude interval mezi naměřenými hodnotami delší.

2.2.5. Matematická analýza

Poslední částí měřícího programu bude sekce nazvaná *Matematická analýza*, která se bude zabírat parametrickou matematickou analýzou jevu nasákavosti vláken. Data, naměřená v experimentu v sekci *Dynamické měření*, se zde budou moci (v omezené míře) upravovat a poté program nalezne metodou nejmenších čtverců parametry známého matematického modelu jevu, které zobrazí v příslušných polích programu [1], [2].

3. Implementace programu

Program, určený pro komunikaci s váhami po sériové lince je vytvořen a zkompilován ve vývojovém prostředí Delphi 5.0 od firmy Inprise (dříve Borland) a je navržen tak, aby mohl být bez problémů používán i na starších počítačích s omezenými systémovými zdroji.

Program se skládá ze tří na sobě nezávislých částí, které se nacházejí v hlavním okně aplikace, mezi jednotlivými částmi se uživatel může přepínat pomocí záložek ve spodní části hlavního okna. Ke komunikaci s digitálními váhami firmy KERN řada 770 slouží sériové rozhraní RS-232-C, kterému patří následující kapitola.

3.1. Sériové rozhraní RS-232-C

Sériové rozhraní RS-232-C (standardizované Electronic Industries Association EIA) bylo původně určeno ke spojení koncového datového zařízení (Data Terminal Equipment DTE – např. terminálu nebo počítače) s komunikačním datovým zařízením (Data Communication Equipment DCE - modemem). V průběhu doby se rozhraní RS-232-C začalo používat i mimo obor telekomunikační techniky. V současnosti má přes všechny jeho nedostatky velmi výrazné postavení i v měřicí technice. Je aplikováno zejména u levnějších měřících přístrojů, speciálních měřících modulů, inteligentních snímačů a plotterů.

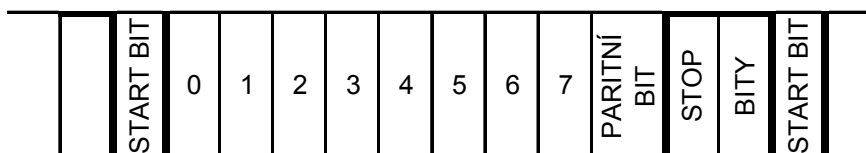
3.1.1. Elektrické parametry

V případě RS-232-C odpovídá logická 1 napětíové úrovni -3 až -15 V, logická 0 úrovni $+3$ až $+15$ V. Obvody rozhraní jsou nesymetrické, proto se uvedené úrovně vztahují vůči potenciálu nulového signálového vodiče. Odpor zátěže se může pohybovat v rozmezí 3 až $7\text{ k}\Omega$, kapacita zátěže nesmí být větší než 2500 pF .

3.1.2. Signály rozhraní a parametry přenosu

Standard EIA definuje celkem 20 signálů rozhraní a přiřazuje je konkrétním pozicím na konektoru s 25 kontakty. V případě měřících přístrojů a osobních počítačů se používá pouze 9 základních signálů, jejichž přehled je uveden v tabulce 2.1. Signál *Ring Indicator (RI)* se ale využívá již pouze výjimečně.

Při asynchronním přenosu je naprosto nezbytné shodně nastavit formát přenosu dat a přenosovou rychlost u obou komunikačních zařízení. Formát přenosu dat se skládá z 1 start bitu, 5 až 8 datových bitů (obvykle 7 nebo 8), z maximálně 1 paritního bitu a 1 nebo 2 stop bitů. Formát přenosu je zobrazen na obrázku 2.1. Přenosová rychlost se volí z řady 110, 300, 600, 1200, 2400, 4800, 9600 nebo 19200 *bit/s*.



Obrázek 3.1. Formát přenosu dat

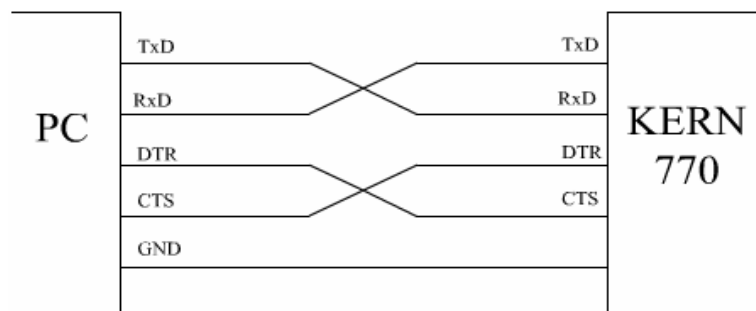
3.2.3. Konkrétní zapojení rozhraní u vah KERN 770

Softwarově řízený přenos dat (software handshaking) rozhraní RS-232-C je možný již u třívodičového zapojení (použito TxD a RxD). Často využívaná metoda (kterou podporují i váhy) je tzv. protokol Xon/Xoff, kdy přístroj přijímající data ovládá komunikaci vysíláním znaků Xon (ASCII 19) a Xoff (ASCII 17). Přístroj na druhé straně tyto znaky rozlišuje od ostatních přijímaných znaků a řídí podle nich komunikaci.

Vysláním znaku Xon signalizuje příjemce, že je schopen data přijímat a druhý přístroj začne s vysíláním, znakem Xoff příjemce přeruší komunikaci. Tato metoda má nevýhodu v tom, že umožňuje přenášet pouze data kódovaná v ASCII kódu.

Hardwarově řízený přenos dat umožňují pěti či sedmivodičová zapojení, obvykle se používají buď dvojice signálů RTS a CTS, nebo DTR a DSR, méně často kombinaci se signálem DCD.

Pro komunikaci s vahami KERN jsem nakonec použil pětivodičové zapojení, které je znázorněno na obrázku 2.2. Použité je hardwarové řízení toku dat s pomocí signálů DTR a CTS. Význam jednotlivých signálů rozhraní RS-232-C je popsán v tabulce 3.1.



Obrázek 3.2. 5-vodičové zapojení RS-232-C

Tabulka 3.1. Nejdůležitější signály rozhraní RS-232-C

signál	číslo kontaktu konektoru			funkce
	symbol	25 kontaktů	9 kontaktů	
Protective Ground	-	1	-	ochranný zemní vodič
Transmitted Data	TxD	2	3	data vysílaná z DTE
Received Data	RxD	3	2	data přijímaná do DTE
Request To Send	RTS	4	7	signál vysílaný z DTE, sděluje DCE, že DTE je připraveno přijímat data
Clear To Send	CTS	5	8	signál vysílaný z DCE, sděluje DTE, že DCE je připraveno přijímat data DTE
Data Set Read	DSR	6	6	signál vysílaný z DCE, sděluje DTE, že DCE je připraveno komunikovat
Signal Ground	-	7	5	signálový zemní vodič
Data Carrier Detected	DCD	8	1	signál vyslaný z DCE, sděluje DTE, že byl detekován signál
Data Terminal Ready	DTR	20	4	signál vyslaný z DTE sděluje DCE, že DTE je funkční
Ring Indicator	RI	22	9	vyzváněcí signál

3.2.4. Programování RS-232-C pomocí Windows API

Pro programování rozhraní mezi PC a vahami KERN jsem použil funkce Windows API, a to konkrétně:

HANDLE CreateFile(LPCTSTR lpFileName, DWORD dwDesiredAccess, DWORD dwShareMode, LPSECURITY_ATTRIBUTES lpSecurityAttributes, DWORD dwCreationDistribution, DWORD dwFlagsAndAttributes, HANDLE hTemplateFile) – funkce otvírající objekt, v našem případě sériový komunikační kanál, a vrací handle, který se dále používá k přístupu k objektu.

BOOL GetCommState(HANDLE hFile, LPDCB lpDCB); – funkce, která naplní strukturu typu DCB (viz níže) stávajícím nastavením komunikačního kanálu COM (vyjádřeno proměnnou *hFile*).

DCB – struktura definující nastavení zařízení. Obsahuje všechny důležité vlastnosti zařízení, jako je parita, počet stopbitů a přenosová rychlost. Tím reprezentuje nastavení sériového kanálu.

BOOL SetCommState(HANDLE hFile, LPDCB lpDCB); – funkce nastavující a inicializující komunikační kanál podle předané struktury typu DCB.

BOOL SetupComm(HANDLE hFile, DWORD dwInQueue, DWORD dwOutQueue) – funkce nastavující parametry přenosu, jako je vstupní a výstupní buffer, pro komunikační kanál.

BOOL CloseHandle(HANDLE hObject); – funkce, která uzavře handle a ukončí práci s objektem – je volána při ukončení komunikace po sériové lince.

Pro čtení dat z kanálu a zápis dat do kanálu se používají funkce pro zápis a čtení ze souboru *WriteFile* a *ReadFile*.

BOOL WriteFile(HANDLE hFile, LPCVOID lpBuffer, DWORD nNumberOfBytesToWrite, LPDWORD lpNumberOfBytesWritten, LPOVERLAPPED lpOverlapped) – funkce zapisující data do objektu zastupovaného handlem, v tomto případě do sériového komunikačního kanálu.

BOOL ReadFile(HANDLE hFile, LPVOID lpBuffer, DWORD nNumberOfBytesToRead, LPDWORD lpNumberOfBytesRead, LPOVERLAPPED lpOverlapped) – funkce čtoucí data z objektu, v našem případě sériového komunikačního kanálu.

BOOL ClearCommError(HANDLE hFile, LPDWORD lpErrors, LPCOMSTAT lpStat) – funkce, která vrací informace o poslední chybě a statusu komunikačního kanálu, zároveň maže příznak chyby kanálu a tím umožňuje další používání kanálu.

Všechny funkce vrací při chybě nenulové hodnoty, bližší informace o typu chyby vrací funkce *DWORD GetLastError(VOID)*.

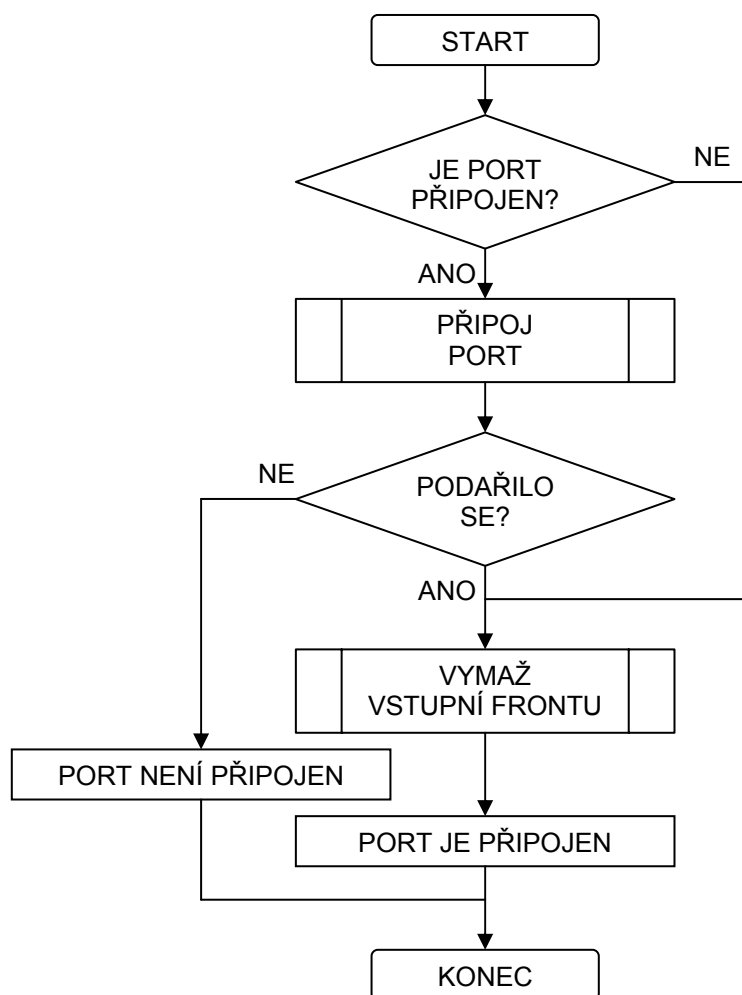
Princip programování komunikace sériovým portem tímto způsobem z hlediska Windows API je jednoduchý. V podstatě se příslušný port otevře jako soubor, poté se nastaví potřebné parametry přenosu a standardními funkcemi *ReadFile* a *WriteFile* se zapisuje a čte do datového toku. Chyby přenosu jsou zjistitelné funkcí *ClearCommError*.

3.2.5. Procedura připojení sériového portu

V programu, ovládajícím digitální váhy jsem použil funkce Windows API popsané v předchozím textu. Procedura, zajišťující připojení sériového portu, zajišťuje též vyprázdnění vstupní fronty portu tak, aby bylo možno ihned zahájit odečítání hodnot. Protože je tato rutinka volána z více částí programu, na začátku zjistí kontrolou globální proměnné *Fcom*, zda je port již připojen a podle toho se buď pokusí port připojit nebo vyprázdní vstupní frontu portu. Všechny tři uživatelem nastavitelné vlastnosti přenosu (jméno portu, přenosová rychlost a parita) jsou proceduře předávány hodnotou, procedura nastavuje globální proměnné *Fcom* a *Hcom* a přepisuje obsah stavového řádku hlavního okna aplikace.

Pro připojení portu se procedura nejdříve pokusí otevřít sériový kanál s pomocí funkce *CreateFile()*, v případě úspěchu uloží s pomocí funkce *GetCommState()* do struktury typu *TDcb*, která reprezentuje nastavení sériového portu, aktuální hodnotu registru příslušného sériového kanálu. Poté hodnotu struktury změní podle požadovaných vlastností spojení a strukturu zkopíruje pomocí funkce *SetCommState()* do registru příslušného sériového portu. Nakonec se funkcí *SetupComm* inicializují

parametry komunikace a zvolí se velikosti vstupního a výstupního bufferu, v tomto případě na 2048 *bytů*.



Obrázek 3.3. Vývojový diagram připojení sériového portu.

Dále procedura smaže vstupní buffer portu a připraví ho funkcí *ClearCommError()* ke komunikaci. Podle úspěšnosti připojení, která je zde reprezentována globální proměnnou *FCom* typu *Boolean*, se upraví text stavového řádku aplikace.

Připojení sériového portu může selhat na několika typech výjimek, z nichž jsou nejčastější výjimky typu:

- Špatně zadaný název portu (systém akceptuje pouze názvy typu *COM1* – *COM9*).
- Pokus o připojení portu, spravovaném jinou aplikací.
- Pokus o zadání nerealistických hodnot nastavení kanálu.

Připojení portu neselže, pokud je odpojen kabel vah, nebo jsou parametry přenosu nastaveny pro obě zařízení různě. V tomto případě bude port otevřen, ale program buď nebude schopen číst data anebo bude číst nepřesně.

Výpis celé této procedury je uveden v příloze B.1.

3.2.6. Odpojení sériového portu

Aplikace ruší připojení sériového portu ve dvou případech. Pokud uživatel mění nastavení portu, pak aplikace odpojí port a opět jej připojí již s novým nastavením.

Druhá situace, kdy aplikace odpojuje sériový kanál, se naskýtá na konci běhu programu. Systém Windows by měl odpojení portu na konci hlavního procesu aplikace zajišťovat sám stejně jako rušení dalších zdrojů, např. alokace paměti nebo přístup do souborů. Často tomu tak ale nebývá, zvláště v případě starších systémů (Windows 95 a 98), které nejsou založeny na jádru NT. Odpojením kanálu na konci programu se předchází stavu, kdy systém neuvolní daný komunikační kanál a ten nelze až do dalšího restartu systému používat žádným jiným procesem.

Proto událost *TForm1.FormClose()* volá funkci *CloseHandle()* s parametrem *HCom*, což je globální proměnná používaná pro přístup k portu, která port uzavře a zpřístupní ho ostatním procesům.

3.2.7. Čtení dat ze sériového kanálu

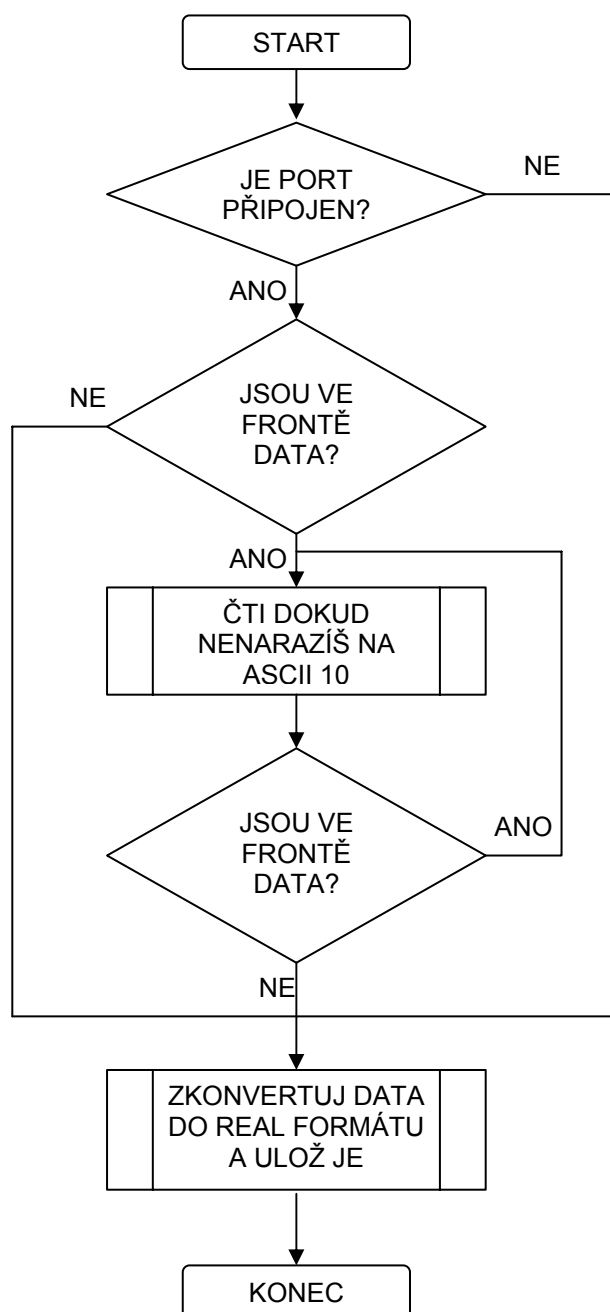
Pro časování sběru dat ze sériového kanálu v části aplikace *Statické měření* i v části *Dynamické měření* slouží standardní komponenta Delphi *TTimer*, konkrétně *Timer1* respektive *Timer2*.

Ke sběru dat jsem použil dvě různé komponenty *TTimer* z toho důvodu, že každá část programu ukládá naměřená data jiným způsobem.

Statické měření zobrazuje poslední získanou hodnotu hmotnosti na displeji, získaného úpravou komponenty typu *TPanel*. *Dynamické měření* přidává data do grafu naměřených hodnot a zároveň do seznamu naměřených hodnot. Také sleduje dobu, která uplynula od spuštění dynamického měření, a pokud délka měření překročí velikost intervalu zadanou uživatelem, tak měření přeruší a uloží data.

Zdrojový kód události *OnTimer* časovače *Timer1*, který používá sekce *Statické měření*, je uveden v příloze B.2. Tato událost je spouštěna po uplynutí příslušného časového intervalu časovače, jehož hodnota je uložena ve vlastnosti časovače *TimerX.Interval*, kde *TimerX* je název časovací komponenty.

Kód události nejdříve testem globální proměnné *FCom* zjistí, zda je port připojen, poté pomocí funkce Windows API *ClearCommError()* zkontroluje, zda jsou ve vstupním bufferu nějaká data. Pokud tam nejsou, program ukončí proceduru, pokud se ve vstupním bufferu nějaká data nacházejí, pak začne vlastní čtení dat ze sériového kanálu.



Obrázek 3.4. Vývojový diagram sběru dat ze sériového portu.

Protože data z vah přicházejí nepřetržitě ve formátu popsaném v kapitole 2.1., kde jsou jednotlivé naměřené hodnoty hmotnosti od sebe odděleny znaky ASCII 13 a ASCII

10 (Carriage Return a Line Feed, což je v podstatě konec řádku), rutinka kontroluje přítomnost znaku ASCII 10. Tím načte celý řádek hodnoty. Pokud i po načtení této hodnoty se ve vstupní frontě nacházejí data, znamená to, že načtená data nejsou aktuálními daty, které vyslaly váhy. Data se tedy zahodí a pokračuje se ve čtení portu.

Tímto postupem jsem také elegantně odstranil problém, který v prvních verzích programu čas od času nastal. Pokud totiž událost časovače přišla přesně v době, kdy váhy posílaly na port svá data (dělají to desetkrát za vteřinu), tak tato rutinka přečetla jen část dat a poslala je ke konverzi na číselný formát. Funkce, starající se o konverzi, toto torzo dat označila za nepřeložitelná a data byla zbytečně zahozena. Ve finální verzi programu rutinka čeká na poslední byte dat přichozí z vah, čímž se tato chyba odstraní.

V případě, že ve se vstupní frontě sériového portu nenacházejí již žádná data, procedura zkonvertuje hodnotu na formát plovoucí desetinné čárky, jehož implementaci je věnována následující kapitola.

Událost *OnTimer* objektu *Timer1* zkonvertovanou hodnotu přímo zobrazuje na displeji hlavního panelu aplikace, pokud je v aplikaci zaškrtnuto políčko nazvané *Nepřetržitě ukládat data*, tak se hodnota navíc automaticky přidá k seznamu naměřených hodnot. Událost objektu *Timer2* hodnotu přidává do grafu i do seznamu naměřených hodnot na stránce sekce *Dynamické měření*.

3.2.8. Konverze načtených dat do formátu s plovoucí desetinnou čárkou

Naměřená hodnota, přečtená událostí časovače ze sériového portu, má formát zobrazený v tabulce 2.1. Hodnota je dlouhá 16 bytů, první byte je znaménko, následuje devět bytů vlastní hodnoty, jedenáctý byte je vždy mezera, dvanáctý až čtrnáctý byte reprezentuje jednotku hmotnosti např. gramy. V následující tabulce je několik příkladů dat, která přicházejí z vah na sériový port. Prázdná políčka reprezentují mezeru.

Tab. 3.2. Příklady dat, poskytnutých váhami.

-			2	.	2	3	1	2			g			CR	LF
		2	3	.	0	0	2	2			g			CR	LF
		7	5	.	1	5	4	5			g			CR	LF

Funkce, starající se o konverzi, nalezne pomocí standardní pascalské funkce *Pos()* pozici desetinné tečky ve vstupním řetězci a rozdělí tím část řetězce, reprezentovanou byty 2 až 9, do dvou částí na celou část a desetinnou část. Z těchto dvou částí pak

funkce složí číslo ve formátu s plovoucí desetinnou čárkou a vrátí jej volající proceduře. [1], [2], [3], [5], [6], [7]

Kód této funkce je uveden v příloze B.4.

3.2. Uživatelské rozhraní programu

Aplikace *Kern* se skládá z jednoho hlavního okna a dialogu nastavení konfigurace sériové linky. Hlavní okno obsahuje tři části, které jsou umístěné na standardní komponentě typu *TPageControl*. Mezi jednotlivými částmi se lze přepínat pomocí záložek této vizuální komponenty.

3.2.1. Spuštění aplikace

Po spuštění aplikace a vytvoření dialogových oken je nutno přechíst inicializační soubor, v kterém je uloženo poslední zadané nastavení sériového portu a pokusit se podle těchto údajů připojit daný sériový port.

Informace o konfiguraci aplikace jsou uloženy v *INI* souboru, jehož formát umožňuje jednoduché ukládání a čtení dat až do velikosti souboru 64kB. Soubory s příponou *.ini jsou uloženy povětšinou v systémovém adresáři Windows, ale mohou se uložit i spolu s aplikací, což umožňuje jednodušší přenos aplikace mezi počítači nebo systémy. Tento formát byl po příchodu 32-bitového systému (Microsoft Windows 95) nahrazen systémem tzv. registrů, které jsou uloženy v jednom souboru a o jejichž správu se nestará obslužná aplikace, ale jádro systému. Firma Microsoft doporučuje používání registrů, ale pro jednodušší aplikace je stále výhodnější a transparentnější využívání *INI* souborů.

Inprise Delphi má ke správě *INI* souborů výkonnou knihovnu nástrojů, která zahrnuje funkce pro přidávání, odebírání a čtení informací ze souboru a také vytváření a rušení souborů.

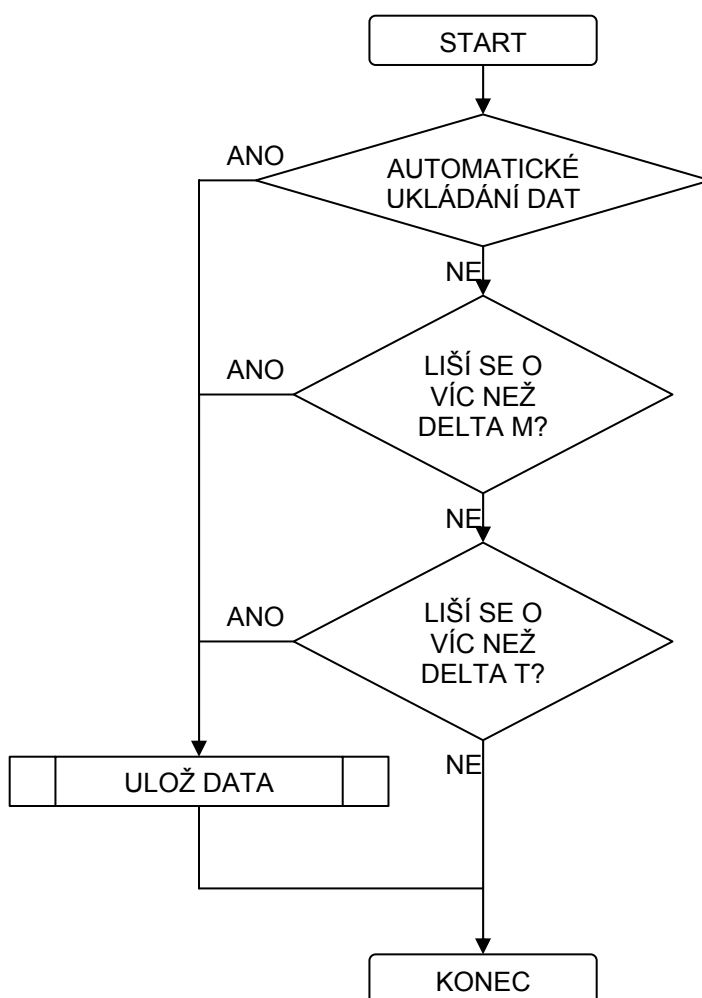
Načtení konfigurace programu probíhá v událostní proceduře *OnCreate* hlavního okna programu *Form1*. Po načtení informací o konfiguraci se procedura voláním funkce *PripojPort()* pokusí připojit port a nastavit vlastnosti přenosu.

3.2.2. Část programu *Statické měření*

Část programu nazvaná *Statické měření* slouží k zobrazování právě naměřených hodnot na displeji, zapisování těchto hodnot do seznamu naměřených hodnot a ukládání tohoto seznamu. Tato část aplikace se skládá z těchto hlavních částí:

- Displej, zobrazující naměřenou hodnotu, je získán ze standardní komponenty *TPanel*. Při aktivním propojení aplikace s váhami zobrazuje aktuální hodnotu, naměřenou váhami. Jeho funkce je totožná s hlavním LCD displejem vah. Hodnotu zobrazenou v tomto panelu je možno pomocí dialogového tlačítka *Zaznamenat* přidat do seznamu hodnot.
- Seznam hodnot je vytvořen z komponenty *TStringGrid*. Kód, přidávající data do seznamu, je vypsán v příloze B.5.
- Dialogové tlačítko *Uložit Seznam* slouží k uložení seznamu hodnot v textovém formátu, přičemž jednotlivá data v řádku seznamu jsou od sebe oddělena tabulátorem. Tento formát je zvolen proto, že se data v tomto formátu dají jednoduše využívat mnoha programy jako je Lotus, Excel, různé databázové systémy nebo Matlab.

3.2.3. Část programu *Dynamické měření*



Obrázek 3.5. Vývojový diagram algoritmu dynamického ukládání dat

Dynamické měření slouží k automatickému proměřování časově proměnného jevu. Je určeno pro laboratorní měření jevů vzlínivosti kapaliny na Katedře zušlechťování materiálů Textilní fakulty Technické univerzity Liberec, ale využití má podstatně širší. Je použitelné pro jakékoliv časově proměnné jevy měřitelné na váhách Kern řady 770.

Okno *Dynamického měření* se skládá z těchto částí:

- Displeje a seznamu hodnot, které mají v podstatě stejnou funkci jako v sekci *Statické měření* s tím rozdílem, že do seznamu hodnot se data ukládají automaticky vždy po určeném časovém intervalu (viz níže).
- Grafu naměřených hodnot. Tato komponenta (*TChart*) je standardní část balíku, dodávaného se systémem Inprise Delphi, a umožňuje jednoduchým způsobem vytvářet průběhy hodnot, sloupcové grafy, speciální ekonomické průběhy atd. Klíčová metoda tohoto objektu pro přidávání dat do průběhu je metoda *Add()* respektive *AddXY()*. Při použití první metody není možno specifikovat vodorovnou souřadnici pro přidávanou hodnotu. Proto jsou data přidávána automaticky a vodorovná osa představuje posloupnost zadaných údajů. Pro dynamické měření jsem v programu použil druhou metodu a na vodorovnou osu vynáším čas v sekundách od začátku měření (od stisku tlačítka *Start*).

Komponenta nativně podporuje posun zobrazeného průběhu a jeho zvětšování myší, bohužel už nepodporuje návrat k původnímu zvětšení ani zmenšování zobrazené oblasti. Proto jsem do levého dolního rohu komponenty přidal jednoduché obrázkové dialogové tlačítko, které po stisku volá metodu komponenty *TChart UndoZoom()*.

- Další část *Dynamického měření* umožňuje nastavování a řízení vlastního procesu měření. Skládá se z editačních boxů, do kterých se zadávají parametry měření a tlačítek, kterými lze v reálném čase řídit průběh měření.

Časový interval, po kterém se hodnoty ukládají do seznamu hodnot a zároveň přidávají do grafu, může uživatel zadat dvěma způsoby. Může pevně nastavit tento interval v sekundách nebo použít automatické nastavení intervalu, kdy proces sleduje měřenou veličinu a data uloží jen v případě, kdy se od předchozí uložené hodnoty liší o zadanou hmotnost. V tomto případě je u stabilnějších průběhů možné, že se po delší čas (nebo dokonce za celou dobu měření) žádná hodnota neuloží a vznikají tak nežádoucí proluky v naměřeném průběhu. Proto je automaticky nastavitelný interval doplněn ještě o editační box, kde se definuje maximální interval, po kterém se hodnota uloží do seznamu naměřených hodnot i bez splnění podmínky změny hmotnosti.

3.2.4. Matematická analýza

V poslední části programu, nazvané *Matematická analýza*, je řešeno nalezení parametrů diferenciální rovnice, popisující dynamiku nasákavosti vláken. Matematický model dynamiky nasákavosti vláken je popsán touto diferenciální rovnicí:

$$\frac{dm_T}{dt} = \frac{R^2 \cdot \rho \cdot g \cdot (m_{MAX} - m_T)}{8 \cdot \eta \cdot m_T \cdot k} \quad (1)$$

kde:

- R je průměrný poloměr vláken ve svazku,
- ρ je hustota kapaliny, která smáčí vlákna,
- g je gravitační zrychlení,
- k je korekční koeficient,
- η je dynamická viskozita použité kapaliny,
- m_T je hmotnost nasáklé kapaliny v čase T a
- m_{MAX} je maximální hmotnost kapaliny nasáklé kapilárami.

Analytické řešení této rovnice je:

$$m_T - m_{MAX} \cdot \ln \frac{m_{MAX}}{m_T - m_{MAX}} = \frac{R^2 \cdot \rho \cdot g \cdot t}{8 \cdot k \cdot \eta} \quad (2)$$

Rovnice (2) je nelineární, proto je nutno ji před použitím linearizovat. To lze u této rovnice v zásadě dvěma způsoby. Odmocninovou aproximací nebo substitucí. V projektu jsem použil obě dvě metody.

Linearizace substitucí

Protože se v pravé části rovnice nevyskytuje kromě času žádná časově proměnná veličina a na levé straně rovnice se neobjevuje čas, můžeme levou stranu rovnice nahradit proměnnou, kterou jsem nazval x .

$$x = m_T - m_{MAX} \cdot \ln \frac{m_{MAX}}{m_T - m_{MAX}} \quad (3)$$

Po úpravě dostaneme linearizovanou rovnici, na jejíž levé straně máme novou proměnnou x :

$$x = \frac{R^2 \cdot \rho \cdot g}{8 \cdot k \cdot \eta} \cdot t \quad (4)$$

Tuto rovnici již můžeme dále použít při matematické analýze.

Odmocninová linearizace

Odmocninovou aproximace, která umožňuje analytické vyjádření hmotnosti m je:

$$m_T = \sqrt{\frac{R^2 \cdot \rho \cdot g \cdot m_{MAX} \cdot t}{4 \cdot k \cdot \eta}} \quad (5)$$

Tato rovnice jsem posléze umocněním a substitucí $y = m_T^2$ linearizoval:

$$y = \frac{R^2 \cdot \rho \cdot g \cdot m_{MAX}}{4 \cdot k \cdot \eta} \cdot t \quad (6)$$

Nevýhoda této aproximace dlí v tom, že je to pouze nepřesné přiblížení ke skutečnému časovému průběhu změny hmotnosti nasáklé tekutiny. Nyní je možno přistoupit pomocí rovnic (4) a (6) k vlastnímu hledání parametrů skutečného průběhu metodou nejmenších čtverců.

Hledané parametry jsou tady dva a to průměrný poloměr vláken ve svazku R a maximální dosažená hmotnost vsáklé kapaliny ve vláknech m_{MAX} . Maximální dosažená hmotnost je neznámou proto, že děj nasákavosti vláken je dlouhotrvající (devadesát procent hmotnosti se dosahuje během dvaceti až čtyřiceti minut podle typu kapaliny a vlákna) a není praktické při měření čekat na dosažení oné maximální hmotnosti a dokonalé ustálení systému. Hodnota hmotnosti se navíc ke své maximální velikosti přibližuje asymptoticky, takže posledních deset procent hmotnosti kapaliny vlákno pojme za třetinovou dobu experimentu. Analýza je tedy navržena tak, že experiment může skončit již při dosažení osmdesát procent maximální hmotnosti vsáklé kapaliny a maximální hodnota je nalezena metodou půlení intervalu.

Princip metody nejmenších čtverců

Metoda nejmenších čtverců se používá pro prokládání empirických, většinou naměřených, hodnot křivkou známého typu a neznámých parametrů. Každá empirická hodnota se od hodnoty proložené funkce v daný čas liší o chybu b . Můžeme tedy napsat, že $y_{E(i)} + b_{(i)} = f_{(x_i)}$. (7)

Chybový parametr b je neznámý, můžeme ale provést odhad, který je založen na nejmenší celkové vzdálenosti proložené křivky od naměřených dat. Vytvoříme odhad součtu kvadrátů vzdáleností S :

$$S = \sum_{i=1}^n [y_i - f(x_i, b_i)]^2 \quad (8)$$

Odhady b_i budou poté určeny z podmínky:

$$S \rightarrow \min \quad (9)$$

Rovnice (9) vyjadřuje princip metody nejmenších čtverců. Pro křivku o známém tvaru lze najít pomocí rovnice (8) a (9) funkci (ve tvaru reziduálního součtu čtverců), kterou se dají určit parametry této funkce.

Přesnost odhadu parametrů funkce je určena právě součtem kvadrátů vzdáleností kolem teoretické regresní (proložené) křivky. Součet kvadrátů je použit při hledání maximální hmotnosti vsáknuté tekutiny do vlákna metodou dělení intervalu.

Oba linearizované průběhy jsou typu $y=bx$ a tedy můžeme podle [4] vypočítat odhad parametru b :

$$b = \frac{\sum_{i=1}^n x_i y_i}{\sum_{i=1}^n x_i^2} \quad (10)$$

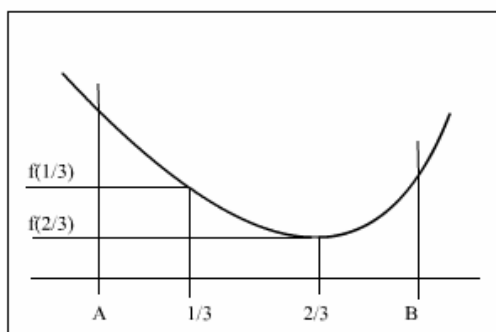
a součet čtverců, podle něhož bude algoritmus hledat hodnotu hmotnosti m_{MAX} , se vypočítá podle vzorce:

$$S_0 = \sum_{i=1}^n y_i^2 - \frac{\left(\sum_{i=1}^n x_i y_i \right)^2}{\sum_{i=1}^n x_i} \quad (11)$$

Nalezení minima funkce metodou dělení intervalu

Protože z výše popsaných důvodů neznáme maximální hmotnost vsáklé tekutiny, jemuž se křivka průběhu asymptoticky přibližuje, musí se tato hodnota nalézt nějakou numerickou metodou.

Přesnost proložení regresní křivkou je dána odhadem rozptylu σ . Čím je σ menší, tím lépe křivka aproximuje daný soubor bodů. Logicky tedy bude nejmenší právě pro m_{MAX} , které je rovno skutečné hmotnosti vsáklé tekutiny. Sledováním odhadu rozptylu σ v závislosti na m_{MAX} a nalezením jeho minima nalezneme i správnou hodnotu m_{MAX} .



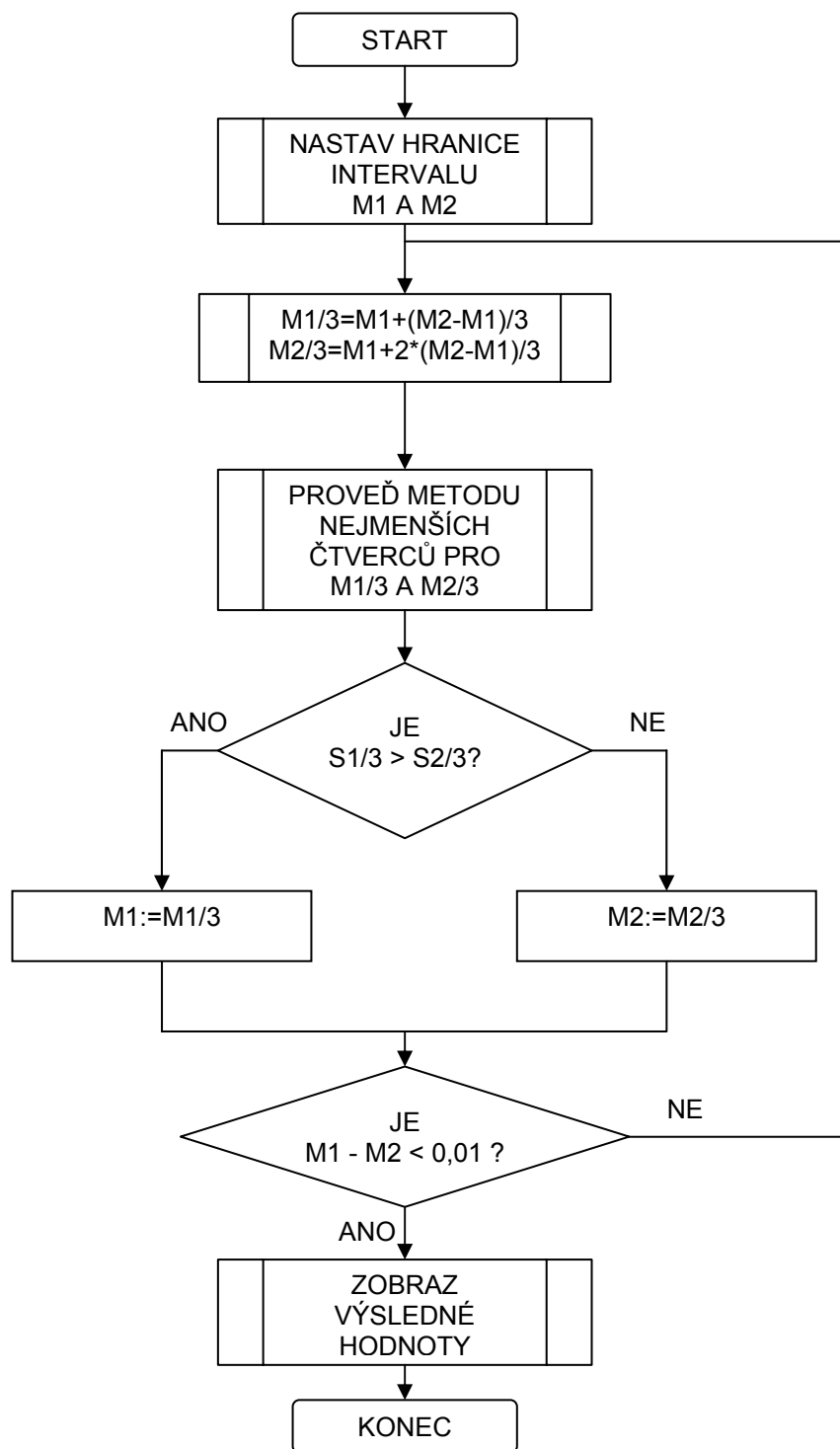
Obrázek 3.6. Metoda dělení intervalu.

Pro nalezení minima této funkce jsem použil metodu dělení intervalu, která porovnává funkční hodnoty v jedné a dvou třetinách intervalu a pro další krok modifikuje hraniční body intervalu podle hodnot funkce ve vnitřních bodech oblasti.

Princip této metody objasním na příkladu, zobrazeném na obrázku 2.6. Původní interval $A-B$ byl rozdělen na třetiny a v bodech $1/3$ a $2/3$ byla zjištěna hodnota funkce f . Protože je hodnota $f(1/3)$ větší než hodnota $f(2/3)$, přesune se pro další krok bod A do bodu $1/3$ a algoritmus se provede znovu. Pokud je velikost intervalu menší než nějaká předem zadaná malá hodnota, algoritmus se ukončí. Tento algoritmus sice najde v daném intervalu pouze jedno minimum, ale protože předpokládáme, že v daném intervalu se nachází pouze jedno minimum, můžeme metodu použít.

Implementace algoritmu

Algoritmus nejdříve zjistí, zda naměřený průběh splňuje podmínku, že poslední hodnota je větší než první hodnota souboru dat, pokud ne tak celý soubor dat vertikálně otočí. Tato část se do programu dostala z důvodu potřeby analyzovat data vzniklá dvěma různými experimenty. První experiment měří přírůstek hmoty kapaliny ve vlákně a druhý měří úbytek kapaliny v kádince. Průběh dat, vzniklý těmito dvěma experimenty je vertikálně obrácený. Odpařování kapaliny z hladiny v nádobce i ze svislé hladiny povrchu vlákna je zanedbáno.



Obrázek 3.7. Vývojový diagram metody dělení intervalu.

Poté je třeba najít na časové ose začátek experimentu, jelikož nejdříve se data nemění, hmotnost kolísá vlivem vnějšího rušení a vibrací okolo počáteční hodnoty. Teprve smočením vlákna v kapalině se prudce změní hmotnost vzorku. Program tedy

porovnává počáteční hodnoty dat a čas začátku experimentu se určí podle první hodnoty, jejíž velikost se od počáteční hodnoty řady dat liší o víc, než je uživatelem zadaná velikost šumu při měření.

Nakonec se provede algoritmus metody dělení intervalu, výsledné hodnoty změny hmotnosti a průměrného průměru vláken se vypíše v hlavním okně programu a provede se grafické proložení dat křivkou. Předpokládáme-li, že měření vzlínivosti skončilo minimálně na 75 procentech z maximální dosažené hmotnosti, maximální hmotnost budeme hledat v intervalu 90 až 130 procent z největší naměřené hodnoty. Výpis částí programu, které implementují metodu dělení intervalu a metodu nejmenších čtverců jsou vypsány v příloze B.

Šířeji se vzlínivosti, jejímu matematickému modelu, experimentům a jejich výsledkům věnuje následující čtvrtá část diplomové práce.

3.2.5. Nastavení programu a tisk

Aby měl uživatel možnost změnit parametry sériového kanálu podle aktuálního nastavení portu vah a používaného portu počítače, přidal jsem do projektu jednoduché modální dialogové okno. Dialog umožňuje nastavit jméno portu (k dispozici jsou porty *COM1* až *COM9*), přenosovou rychlost sériového kanálu a paritu. Možnost nastavení dalších parametrů jako je počet stopbitů nebo typ řízení přenosu dat (softwarový nebo hardwarový) jsem neimplementoval, jelikož změna jejich nastavení není potřebná.

Pro změnu parametrů přenosu sériové linky je třeba připojení portu zrušit, změnit nastavení a znovu port připojit. Toto se uskuteční po stisknutí dialogového tlačítka *OK* na formuláři nastavení. Metoda změny konfigurace ukládá do inicializačního souboru, který je popsán v kapitole 3.2.1.

Z důvodu potřeby tisku oblasti grafu na tiskárně je do třetí části programu, nazvané *Matematická analýza* přidáno tlačítko *Tisk*. Po jeho stisknutí se zobrazí systémový dialog nastavení tiskárny, kde uživatel vybere ze seznamu aktivní tiskárnu, může změnit vlastnosti tisku a po stisknutí dialogového tlačítka *OK* program zavolá metodu objektu *TChart.Print*, která vytiskne aktuálně zobrazovanou oblast grafu na aktivní tiskárně. [1], [2], [3], [4], [5], [6], [7], [8]

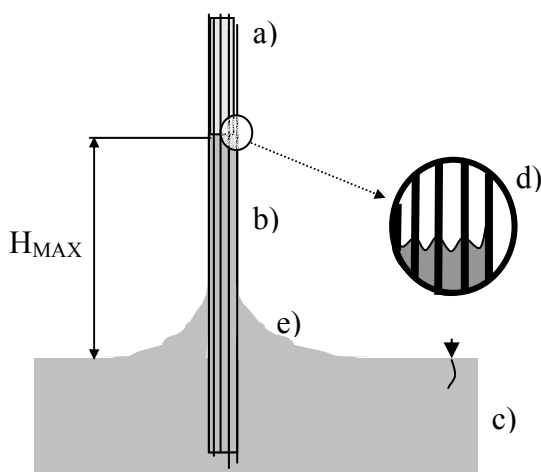
4. Proces vztlínání kapaliny

Řada významných fyzikálních a fyzikálně-chemických procesů je založena na kapilaritě, tedy samovolném pronikání kapaliny do porézních materiálů. Principu kapilarity se využívá v řadě oborů. Většina těchto procesů probíhá v pórech mezi vlákny textilií.

Kapilárním jevům mezi vlákny textilií je věnována v posledních desetiletích značná pozornost, která stále nabývá na intenzitě. Pro potřeby vývoje nových textilních materiálů se studují zejména dynamické kapilární procesy, které jsou úzce spojeny s komfortem uživatele oděvní textilie. Na dynamice vztlínání závisí, jak rychle bude kapalina odváděna z povrchu těla do textilie a jak rychle bude odpařována. Z technických problémů je intenzivně řešen průnik barevné kapaliny do textilie a jiných porézních materiálů při tisku inkoustovými tiskárnami.

4.1. Teoretický úvod

V mé práci jsem sledoval proces pronikání čisté kapaliny do svislého textilního materiálu, jehož vlákna jsou orientována přibližně svisle. Standardní metodou pro sledování dynamiky procesu vztlínání je měření sací výšky.



Obrázek 4.1. Rovnováha vztlínání kapaliny do svislé nitě

Na obrázku 4.1. je znázorněn jev vztlínání kapaliny do svislé nitě, tento jev se skládá z následujících pěti částí:

- a) Svazek rovnoběžných vláken bez kapaliny.

- b) Svazek rovnoběžných vláken s kapalinou, kde maximální výška dosažená kapalinou je označena jako H_{MAX} .
- c) Kapalina.
- d) Detail hladiny kapaliny v úrovni výšky H_{MAX} .
- e) Přejícná oblast mezi smáčenými vlákny a kapalinou.

Základní vztah, kterým se popisuje dynamika vztlínání je téměř sto let starý Lucas-Waschburnův vztah:

$$\frac{dH}{dt} = \frac{R \cdot \rho \cdot g \cdot (2 \cdot \sigma \cdot \cos(\phi) - H_T \cdot R)}{8 \cdot \eta \cdot H} \quad (11)$$

kde:

η - dynamická viskozita kapaliny,

H - sací výška [m] v čase t ,

R - poloměr kapiláry [m],

ρ - hustota kapaliny [$kg \cdot m^{-3}$],

g - gravitační zrychlení [$m \cdot s^{-2}$],

Φ - kontaktní úhel mezi hladinou kapaliny a vláknem [rad] a

σ - povrchové napětí kapaliny [$N \cdot m^{-1}$].

Tento vztah je dostatečně přesnou aproximací kapilarity pouze v nejjednodušších kapilárních systémech. Rovnice vychází z rovnováhy adhezních a gravitačních sil pro kapalinu mezi vlákny a lze ji použít pro výpočet dynamiky vztlínání u systému s rovnoběžnými kapilárami. Pro zkoumání problematiky vztlínání kapaliny do reálné vláknenné struktury je z následujících důvodů téměř nepoužitelný.

- Při vztlínání kapaliny v přízi, nebo jakémkoliv textilním útvaru, hraje roli také hladina kapaliny na svislém povrchu textilie. Tento parametr není v rovnici zahrnut.
- Není možné popsat strukturu textilie jen podle jediného parametru - poloměru pórů. Poloměr pórů se mění s deformací textilní struktury a jeho zdánlivá hodnota bude vysoce závislá na použité kapalině.

Účinek tlaku na rychlost proudění newtonovské kapaliny v trubici popisuje Hagen-Poiseuillův zákon. Objem kapaliny V , který proteče za čas t trubicí o poloměru R a délce l , pokud mezi jejími konci je tlakový rozdíl $\Delta P = (P_2 - P_1)$, je dán výrazem:

$$\frac{dH}{dt} = \frac{R^2 \cdot \Delta P}{8 \cdot \eta \cdot l} \quad (12)$$

Pro popis tlakového spádu je nutné pracovat s hydrostatickým tlakem:

$$P = -H \cdot \rho \cdot g \quad (13)$$

Protože je sledován případ smáčivých povrchů, dochází ke vztlínání nad volnou hladinu kapaliny. Hodnoty H jsou kladné a tlak nabývá záporných hodnot.

Pro případ samovolného vztlínání se mění tlak P a sací výška H a platí:

- ΔP je dána rozdílem mezi hydrostatickými tlaky P_1 a P_2 .
- P_1 je hodnota tlaku, které bude dosaženo na nejvyšším místě vyvzlínané kapaliny v rovnováze – ve výšce H_{max} nad volnou hladinou:

$$P_1 = -H_{MAX} \cdot \rho \cdot g \quad (14)$$

P_2 je aktuální hydrostatický tlak na okamžitém nejvyšším místě vyvzlínaného útvaru kapaliny.

$$P_2 = -H \cdot \rho \cdot g \quad (15)$$

Tlakový rozdíl lze vypočítat jako rozdíl tlaků P_1 a P_2 .

$$\Delta P = P_2 - P_1 = \rho \cdot g \cdot (H_{MAX} - H) \quad (16)$$

“Délka trubice” je přímo úměrná sací výšce H . V ideálním případě přímé kapiláry je délka trubice stejná jako sací výška. V případě zakřivených kapilár (např. v přízi) tomu tak není, proto je vhodné pracovat s korekčním parametrem k :

$$l = H \cdot k \quad (17)$$

Korekční koeficient k obvykle nabývá hodnot 1 – 1,4.

Pak po dosazení (16) a (17) do Hagen-Poiseuillova vztahu (12) dostaneme:

$$\frac{dH}{dt} = \frac{R^2 \cdot \rho \cdot g \cdot (H_{MAX} - H)}{8 \cdot \eta \cdot k \cdot H} \quad (18)$$

Tato rovnice je zobecněním známého Lucas-Waschburnova vztahu – rovnice (11)

Rovnice (18) neumožňuje analyticky vyjádřit závislost sací výšky na čase. Problémy přináší řešení následující rovnice, jejíž pravou stranu nelze analyticky zintegrovat.

$$\int_0^{H(t)} \frac{H}{H_{MAX} - H} dH = \int_0^t \frac{R^2 \cdot \rho \cdot g}{8 \cdot \eta \cdot k} dt \quad (19)$$

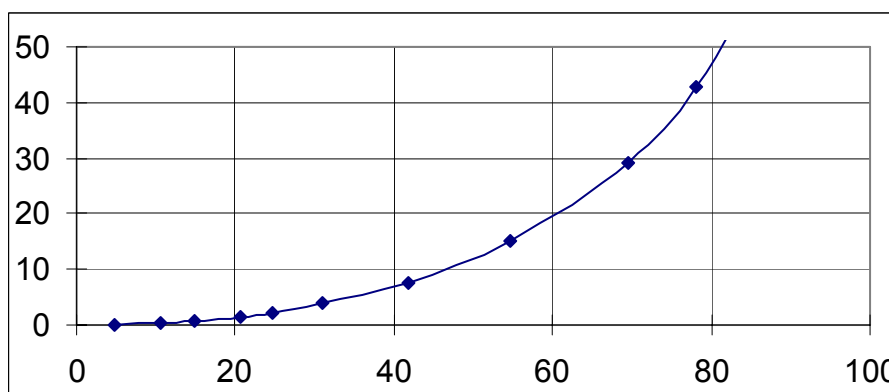
Pro získání analytického řešení je obvykle tento tvar Lucas-Waschburnova vztahu zjednodušen na:

$$\int_0^{H(t)} \frac{H}{H_{MAX}} dH = \int_0^t \frac{R^2 \cdot \rho \cdot g}{8 \cdot \eta \cdot k} dt \quad (20)$$

Řešení pak vede na jednoduchý vztah, který je označován jako odmocninová aproximace:

$$H(t) = \sqrt{\frac{t \cdot R^2 \cdot \rho \cdot g \cdot H_{MAX}}{4 \cdot \eta \cdot k}} \quad (21)$$

Tato rovnice popisuje s vyhovující přesností počáteční část procesu vztlínání. Pokud zvolíme jako kritérium přijatelnosti maximálně 5% odchylku aproximace od reálné hodnoty, pak lze použít odmocninovou aproximaci do dosazení 30% z rovnovážné sací výšky. Nad touto hodnotou chyba odmocninové aproximace prudce stoupá. Tento závěr plyne nejen z regrese numerických dat, ale lze jej pozorovat i na datech experimentálních. O použitelnosti odmocninového modelu si lze udělat představu z následujícího grafu:



Obrázek 4.2. Závislost procentuální chyby odmocninové aproximace na procentech výšky hladiny.

Při použití odmocninové aproximace je tedy pro nalezení parametrů upravené Lucas-Waschburnovy rovnice možno použít pouze dvacet až třicet procent objemu naměřených dat. Proto jsem pro získání parametrů matematického modelu použil také

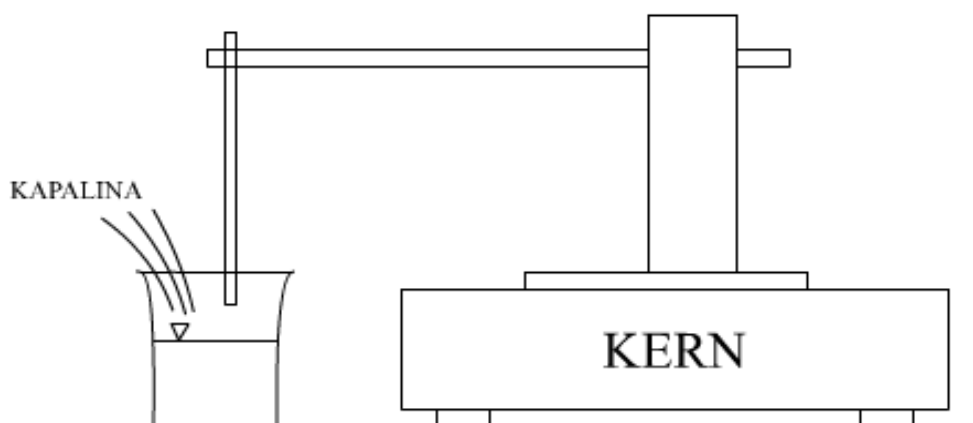
neaproximovaný vztah, ve kterém závislost výšky hladiny (respektive hmotnosti vsáklé kapaliny na čase) není vyjádřena analyticky.[4], [8]

4.2. Experimentální uspořádání

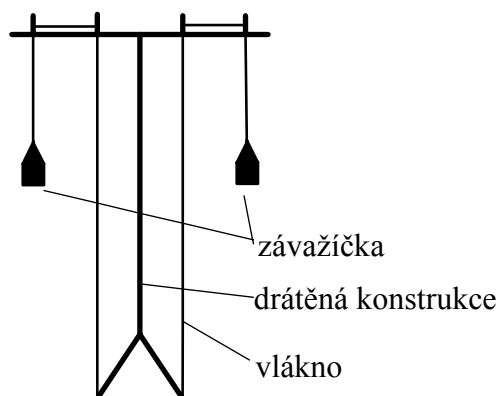
Experiment vztlínání kapaliny do vláknenných materiálů je z důvodů potřeby měření malých změn hmotnosti významně ovlivňován vnějšími podmínkami. Nepatrné vibrace mohou znehodnotit jak několik minut trvající proces měření, tak i měřený vzorek. Kritický je obzvláště okamžik ponoření konce vlákna do kapaliny, což vyvolává vlnění hladiny a vzorek není po prvních několik sekund ve stabilních podmínkách. Rovněž jakákoliv manipulace s měřícím systémem během probíhajícího experimentu je velmi obtížná a často vede k znehodnocení naměřených dat.

Výhody, které s sebou přináší automatizovaný sběr dat, jsou tedy následovány složitostmi, které se musí obtížně řešit. Největší problém je s mikrovibracemi, které vzniknou těsně po ponoření konce vlákna do vody. Dalším problémem je požadavek ponoření co nejmenšího objemu vzorku, jelikož by měření příliš ovlivnila vztlková síla kapaliny. Proto nelze vlákno napínat obvyklým postupem, kdy se na něj zavěsí malé závažíčko

Po několika neúspěšných experimentech jsem navrhl systém, který je zobrazen na obrázku 4.3.



Obrázek 4.3. Experimentální zařízení pro měření vztlínivosti.



Obrázek 4.4. Konstrukce pro napínání vlákna.

Vlákno je napnuto na konstrukci (viz detail na obrázku 4.4.) a na obou stranách je opatřeno závažíčky, která mu poskytují potřebné předpětí. Spodní část konstrukce je vyrobena z velmi slabé struny, čímž se výrazně omezí objem ponořené části přípravku v kapalině.

Konstrukce je postavena na váhy tak, aby těžiště celého systému bylo ve středu vah, umístěných na pevné podložce. Druhý konec konstrukce je umístěn v kádince, do které se opatrně nalévá voda až do okamžiku, kdy se hladina dotkne spodní části konstrukce a vlákna. Kapalina začne ihned pronikat do pórů vlákna, což je vidět na grafu hlavního okna aplikace jako prudký vzestup hmotnosti měřícího přípravku.

Při vlastním experiment jsem použil polyamidovou přízi. Experiment probíhal za laboratorních podmínek při teplotě 23°C , ne při stoprocentní vlhkosti. Experiment byl tedy zatížen chybou, která vzniká při odpařování kapaliny ze svislé hladiny na okraji vlákna. Mým cílem ale nebylo naměřit přesné hodnoty vztlínání, ale ověřit metodu a měřící pracoviště pro použití v podmínkách laboratoří Textilní fakulty. Výsledné grafy experimentu a regresního prokládání křivkou jsou zobrazeny níže. Z grafů je patrné, že použití odmocninové aproximace dobře kopíruje skutečný průběh do přibližně 30% maximální hodnoty hmotnosti, zatímco proložení bez aproximace kopíruje průběh po celé délce. Použití odmocninové aproximace se tedy jeví jako nevýhodné z toho důvodu, že je nutno použít jenom prvních 30% - 40% naměřených dat.

Oba grafy jsou získány přímo tiskem, který je implementován aplikací a který je popsán v kapitole 3.2.5.

4.3. Grafy

4.3.1. Naměřená data proložená s odmocninovou aproximací.

4.3.2. Naměřená data proložená bez aproximace.

5. Závěr

Náplní této práce bylo vytvoření funkční aplikace v systému Microsoft Windows, která je schopná sběru naměřených dat z vah třídy 770 od německé firmy KERN.

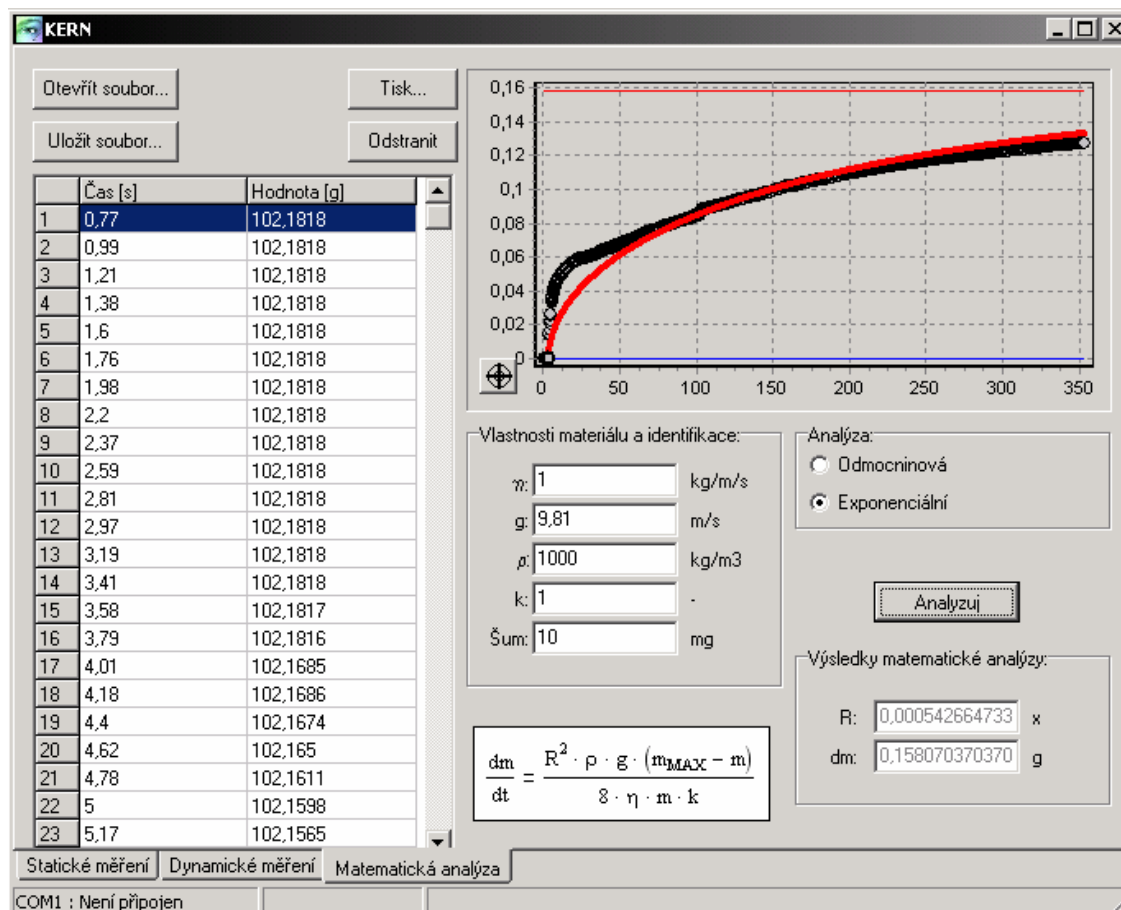
V první části jsem se zajímal o konkrétní styl obsluhy sériového portu těchto digitálních vah, formát vstupních a výstupních dat. Bohužel, systém použitý u vah neumožňuje kalibraci ani změnu nastavení konfigurace vah přes sériový port, a proto není nastavení vah do aplikace implementováno. Jinak váhy umožňují klasickou pětižilovou asynchronní komunikaci s hardwarovým řízením přenosu dat, které jsem tedy použil pouze ke sběru dat.

Vlastní program se skládá ze tří částí, které umožňují zobrazovat údaje o hmotnosti na obrazovce a jejich automatické nebo manuální ukládání. Pozornost si zaslouží zejména část dynamického měření, která je schopna ukládat data rychlostí až pět údajů za sekundu (limitováno rychlostí obnovování dat převodníkem vah), což by nebylo možné uskutečnit neautomatizovaným systémem. Data se pro další zpracování ukládají ve formátu akceptovatelným mnoha různými aplikacemi. Pro jev kapilárního vztlínání kapaliny do textilních materiálů je implementována matematická analýza, která hledá parametry Lucas-Waschburnova vztahu.

Vytvořený program je napsán a zkompilován tak, aby byl plně funkční i na pomalejších počítačích a umožňuje komfortní měření obzvláště déletrvajících procesů a bude jistě přínosem pro laboratorní práci na Textilní fakultě Technické univerzity v Liberci. V budoucnu by aplikace měla pomoci hlavně ve výzkumu již zmíněného jevu kapilární vztlínivosti kapaliny. Aplikace umožní rychlejší a jednodušší měření bez nutnosti barvení kapaliny a složitého odečítání sací výšky a bude možno detailně proměřit i rychlé změny hmotnosti na začátku tohoto jevu.

Příloha A. Uživatelská příručka

Tato příručka pojednává pouze o aplikaci měření, objasňuje práci s vlastním programem, jeho nastavení a princip propojení vah s počítačem.



Obrázek.A.1. Hlavní okno programu.

A.1. Instalace a nastavení programu

Aplikace nevyžaduje instalaci. K správné funkci stačí pouze spustit soubor Kern.exe. Po prvním spuštění aplikace na počítači budete vyzváni k zadání správných parametrů sériové linky tak, aby korespondovaly s nastavením vah.

A.1.1. Nastavení portu RS-232-C vah

Nastavení vah zjistíte ve vlastním menu vah, do kterého se dostanete podržením tlačítka *TARE* při zapínání vah.

Tabulka A.1. Kódy menu pro nastavení sériové linky pro váhy KERN 770.

Přenosová rychlost	Kód	Parita	Kód
150	5 1 1	Mark	5 2 1
300	5 1 2	Space	5 2 2
600	5 1 3	Odd	5 2 3
1200	5 1 4	Even	5 2 4
2400	5 1 5		
4800	5 1 6		
9600	5 1 7		
19200	5 1 8		

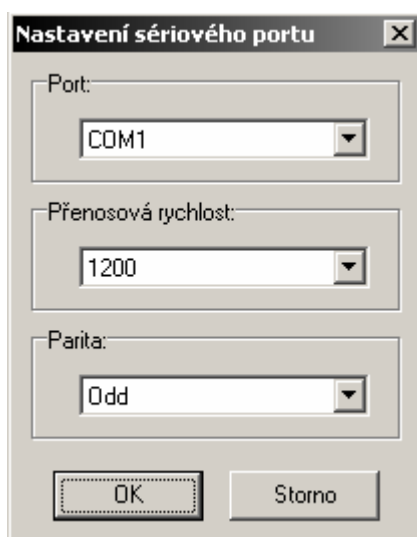
Kódy, potřebné k nastavení sériové linky na straně vah, jsou vypsány v Tabulce A.1. Křížkem je vyznačeno tovární nastavení, které nedoporučuji měnit. K továrnímu nastavení je též možno přistoupit po odeslání kódu 9 – 1.

Přístup do menu

- Tlačítkem *ON/OFF* vypněte váhy.
- Opět zapněte váhy. Ve chvíli, kdy se zobrazí všechny segmenty displeje vah, stiskněte tlačítko *TARE* a uvolněte jej, až když se na displeji objeví číslice 1.
- Jste v menu nastavení vah. Nyní tlačítkem *CAL* vyberte číslici menu, kterou chcete zvolit.
- Tlačítkem *PRINT* svou volbu potvrdíte a kurzor se přesune na další pozici.
- Nakonec nastavení potvrdíte a uložíte podržením tlačítka *TARE* (váhy se restartují).

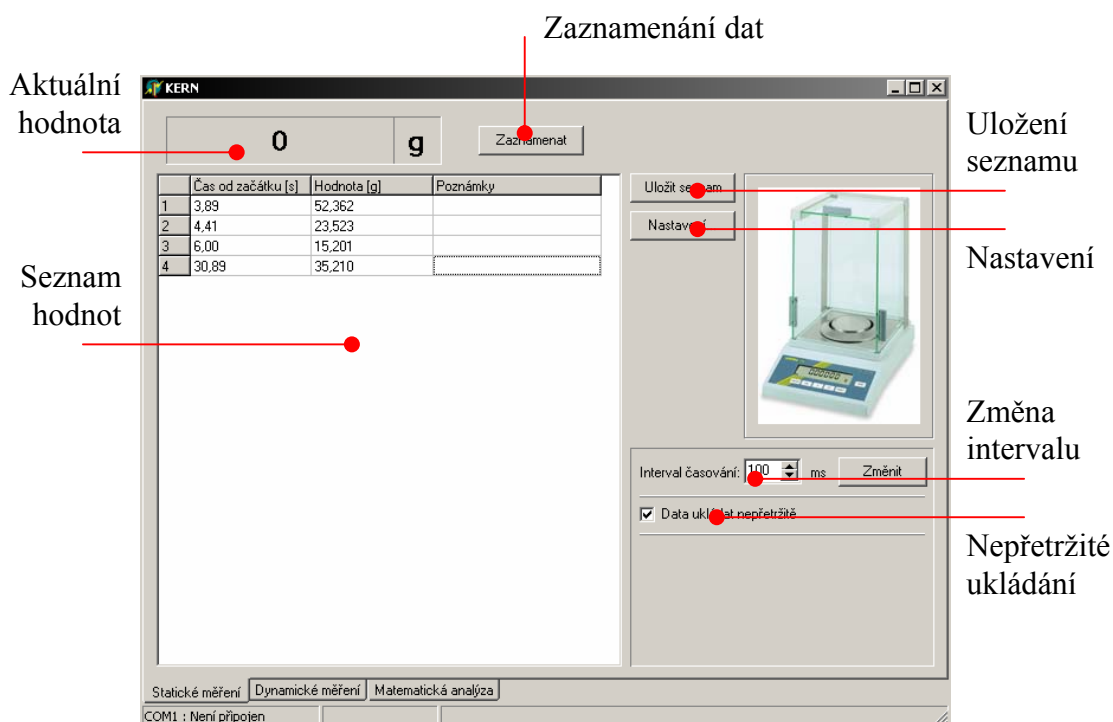
A.1.2. Nastavení parametrů sériové linky programu

Pro přístup k dialogu *Nastavení sériového portu* stiskněte tlačítko *Nastavení* na kartě *Statické měření*. Zobrazí se dialog, ve kterém můžete měnit parametry sériové linky RS-232-C. Stiskem dialogového tlačítka *OK* se provede nastavení programu a zároveň se informace uloží pro další spuštění programu. Tlačítko *Storno* ukončí dialog bez uložení.

Obrázek. A.2. Dialog *Nastavení sériového portu*.

Pro správnou funkci programu musí být obě strany, tedy jak váhy tak samotný program, stejné hodnoty vlastností sériového portu. Z důvodů rychlosti přenosu nedoporučuji používat nižší přenosovou rychlost než 1200 baudů.

A.2. Statické měření



Obrázek A.3. Statické měření.

Po spuštění aplikace *Kern.exe* se otevře okno zobrazené na Obrázku A.3. Toto hlavní okno aplikace je rozděleno na tři části a to *Statické měření*, *Dynamické měření* a *Matematická analýza*. Mezi těmito částmi se lze přepínat s pomocí dialogových tlačítek

ve spodní části okna. Pod těmito tlačítky se dále nachází stavový panel, vypisující informace o spojení počítače s vahami.

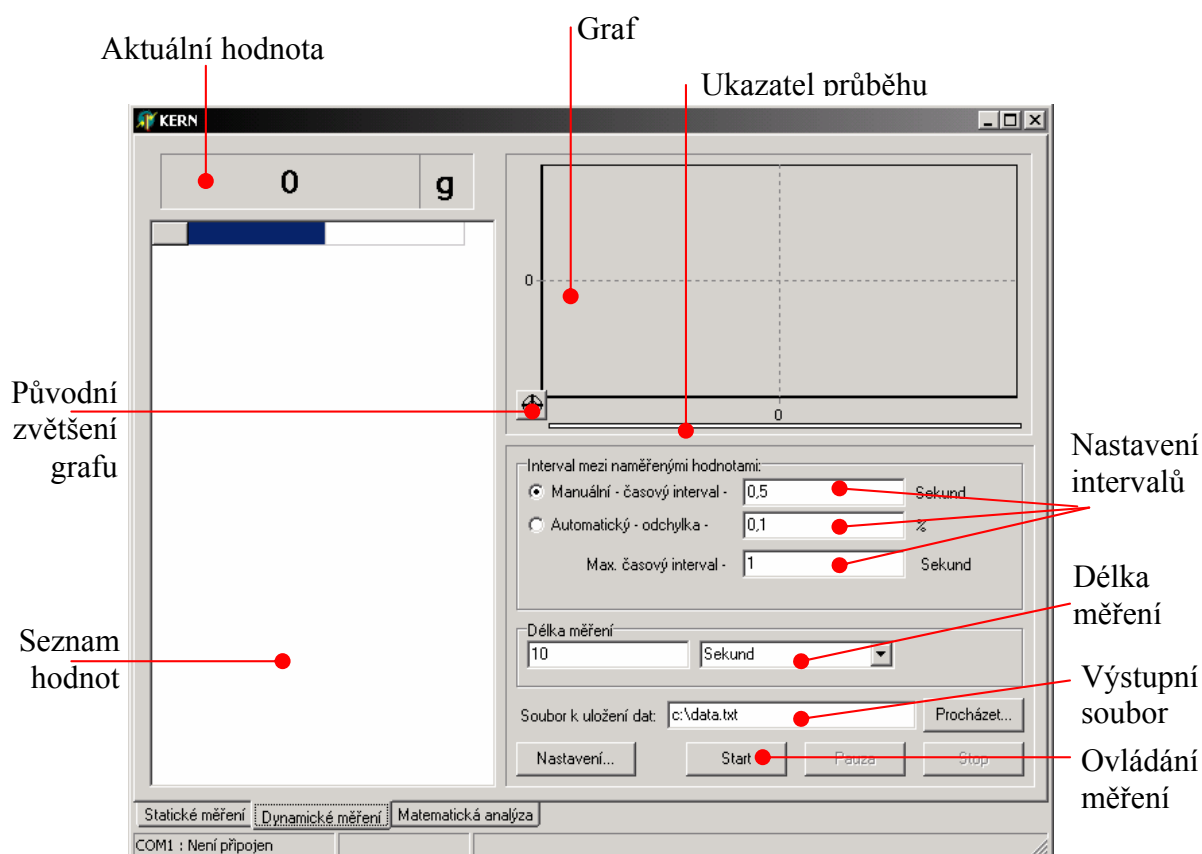
První část programu, *Statické měření*, obsahuje panel, který zobrazuje momentální hodnotu naměřené hmotnosti, dále seznam naměřených hodnot a několik funkčních tlačítek, jejichž funkce je popsána dále. Část statické měření umožňuje přímé měření hmotnosti s ukládáním dat do souboru a dále zapisování poznámek o naměřených hodnotách.

Statické měření obsahuje tyto komponenty:

- *Aktuální hodnota* – panel zobrazuje aktuální hodnotu naměřené hmotnosti. Tato hodnota se dá pomocí tlačítka *Zaznamenat* přenést do seznamu hodnot.
- *Seznam hodnot* – je tabulka naměřených hodnot a obsahuje tyto sloupce:
 - Pořadové číslo měření.
 - Čas měření v sekundách od spuštění aplikace.
 - Naměřená hodnota hmotnosti.
- *Poznámky* – tyto pole se dají využít k zapsání poznámek o naměřených hodnotách nebo o podmínkách měření, jako je datum a čas měření, teplota, tlak, vzdušná vlhkost a podobně.
- Tlačítko *Uložit seznam* slouží k uložení seznamu hodnot v takzvaném tabelátorovém formátu. Tento formát byl zvolen, protože jej podporuje pro import velké množství aplikací, například Microsoft Excel či Matlab od firmy Math Works.
- Tlačítko *Nastavení* otevře dialog nastavení sériové linky, který je zobrazen na Obrázku A.2.
- Sekce, která je na Obrázku A.3 pojmenována jako *Změna intervalu* a skládá se z editačního boxu a tlačítka, nastavuje rychlost obnovování hodnot na panelu *Aktuální hodnota*. Časová hodnota v editačním boxu je vyjádřena v milisekundách.
- Zaškrtnuté políčko *Nepřetržité ukládání* slouží k automatickému přidávání naměřených hodnot do seznamu naměřených hodnot. Pokud je toto políčko zaškrtnuté, plní se seznam naměřených hodnot rychlostí nastavenou v sekci *Změna intervalu*.

A.3. Dynamické měření

Část *Dynamické měření* je specializována na měření v časovém intervalu. Uživatel si může zvolit dobu měření a časové intervaly, po kterých se naměřená data získaná aplikací ukládají do souboru a zobrazují v grafu. Velikost časových intervalů pro ukládání dat se dá zvolit buď přímo nebo nastavením relativní odchylky od poslední naměřené hodnoty. Po jejím překročení se data uloží a zakreslí do grafu. Druhý způsob je výhodný zvláště v případech, kdy naměřená veličina kolísá okolo nějaké ustálené hodnoty, nebo přechází z ustálené hodnoty rychle na jinou hodnotu. Vyhneme se tak zbytečně velkému počtu naměřených hodnot a zároveň dobře proměříme ty části průběhu, které se mění rychle.



Obrázek A.4. Dynamické měření.

K oknu *Dynamické měření* tak, jak je zobrazené na Obrázku A.4, se dostaneme, když vybereme příslušné tlačítko menu na spodu formuláře. *Dynamické měření* obsahuje tyto komponenty:

- *Seznam hodnot* a *Aktuální hodnota* mají stejnou funkci jako ve *Statickém měření*, pouze seznam má jen tři sloupce (pole na poznámky již není potřeba):

- Pořadové číslo měření.
- Čas měření v sekundách od spuštění aplikace.
- Naměřená hodnota hmotnosti.
- Na panelu *Graf* se zobrazuje průběh naměřené hodnoty v závislosti na čase, výběrem obdélníkové oblasti levým tlačítkem myši lze graf zvětšovat, stisknutím a podržením pravého tlačítka myši se lze v grafu pohybovat. Původní pozice a zvětšení grafu se dosáhne stisknutím tlačítka *Původní zvětšení grafu*.
- Komponenty v sekci *Celková délka měření* slouží k nastavení délky měření, po které se měření automaticky ukončí.
- Editační boxy a zaškrtačací políčka v sekci *Nastavení intervalů* slouží k specifikaci časových intervalů, po kterých se ukládají jednotlivé naměřené hodnoty. Intervaly se dají nastavit dvěma způsoby:
 - *Manuální nastavení* – uživatel určuje přímo časové intervaly mezi ukládáním. Tento způsob má jednu nevýhodu a to, že uživatel musí mít dopředu odhadnout rychlost průběhu hmotnosti měřeného jevu.
 - *Automatické nastavení* – je výhodnější v situacích, kdy uživatel nezná přesnou dynamiku měření, nebo naměřená veličina vykazuje velké změny dynamiky. Uživatel nastaví citlivost měření v miligramech a maximální délku časového intervalu. Systém poté sleduje měřenou veličinu a uloží ji, pokud se od poslední uložené hodnoty liší o víc než je nastavená citlivost, nebo pokud časový interval od naposledy uložené hodnoty je vyšší než nastavená maximální délka intervalu. Tento způsob ukládání dat lépe proměřuje rychlejší změny hmotnosti, přičemž neukládá zbytečně velké množství dat tam, kde se měřená hodnota nemění.
- Sekce *Výstupní soubor* slouží k definování jména souboru, do kterého se budou ukládat naměřená data. Soubor má stejný formát jako soubor, do kterého se ukládá seznam dat v části *Statické měření*.
- Tlačítka *Ovládání měření* slouží k spouštění, pozastavování a zastavení vlastního měření.

Jak postupovat při měření dynamického průběhu

1. Ujistěte se, že je dobře nastavená sériová linka a komunikace s vahami – displej *Aktuální hodnota* by měl ukazovat stejnou hodnotu, jako je zobrazena na displeji vah.

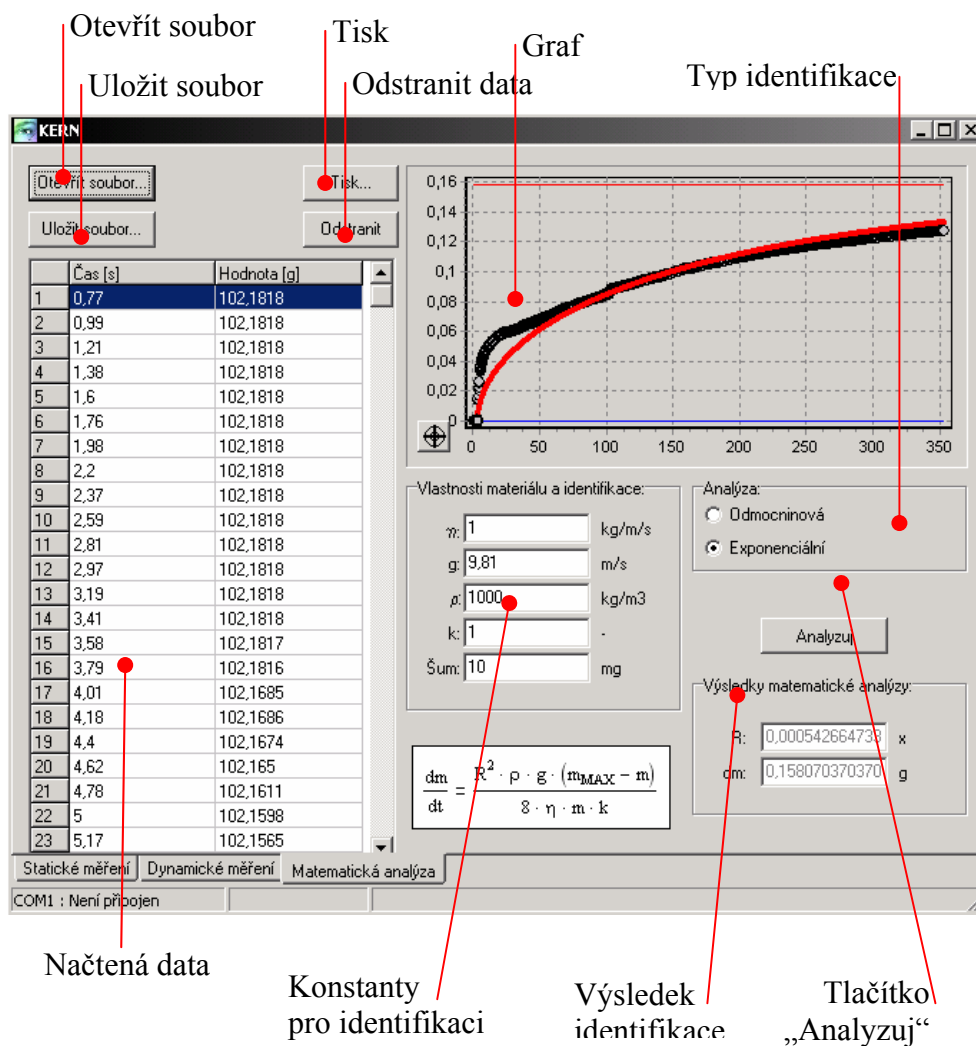
2. Nastavte celkovou dobu měření – je lépe nastavit dobu o trochu větší, než je odhadovaná doba trvání sledovaného jevu, protože měření můžete kdykoliv předčasně ukončit tlačítkem *Stop*.
3. Nastavte požadované časové intervaly mezi uloženými hodnotami.
4. Do editačního boxu napište jméno souboru, do kterého chcete ukládat naměřené hodnoty.
5. Po stisknutí tlačítka *Start* se inicializuje měření a začnou se podle nastavených intervalů ukládat naměřené hodnoty do souboru, zobrazovat se v grafu a seznamu na obrazovce. Měření je ukončeno po uběhnutí dané doby měření, v případě potřeby je možno průběh měření pozastavit či zastavit.

A.4. Matematická analýza

Část programu, nazvaná *Matematická analýza*, je specializována na nalezení parametrů modelu nasákavosti vláken. Diferenciální rovnice modelu je zobrazená ve spodní části dialogového okna. Dialogové okno se skládá z těchto komponent:

- Komponenty *Graf* a tabulka *Načtená data* slouží k zobrazování načtených dat a k zobrazení parametrizovaného průběhu hmotnosti vsáklé tekutiny na čas.
- Tlačítkem *Tisk* lze zobrazit dialog nastavení tiskárny a následně tisknout oblast grafu tak, jak je uvedeno v následující kapitole.
- Tlačítko *Otevřít soubor* slouží k načtení seznamu hodnot z textového souboru do tabulky a následnému zobrazení v grafu. Tlačítko *Uložit soubor* slouží k uložení dat do souboru.
- Tlačítko *Odstranit* slouží k editování naměřených dat. Není povolena změna hodnot, je možno pouze odstranit řádek z tabulky seznamu hodnot. Tento způsob nakládání s daty je dovolen proto, že naměřená data jsou často ovlivňována vnějšími podmínkami a je třeba některé části dat odstranit a vyhladit tak naměřený průběh.
- Do skupiny editačních boxů, která je na obrázku A.5 nazvána jako *Konstanty pro identifikaci* se zadávají vlastnosti tekutiny použité při vlastním pokusu se vztlínavostí. Tyto konstanty jsou poté využívány při parametrické identifikaci.
- Dále se v sekci *Typ identifikace* nastaví, zda se má pro identifikaci použít odmocninový nebo exponenciální typ linearizace diferenciální rovnice systému.

Platí, že odmocninová identifikace prokládá lépe počáteční hodnoty jevu, exponenciální spíše hodnoty, které se přibližují k ustálené hodnotě.



Obrázek A.5. Matematická analýza.

- Tlačítkem *Analyzuj* se spustí vlastní parametrizace. Program provede identifikaci hodnot, následně vykreslí do grafu průběh parametrizovaného jevu a do sekce *Výsledek identifikace* vypíše vypočítané hodnoty.

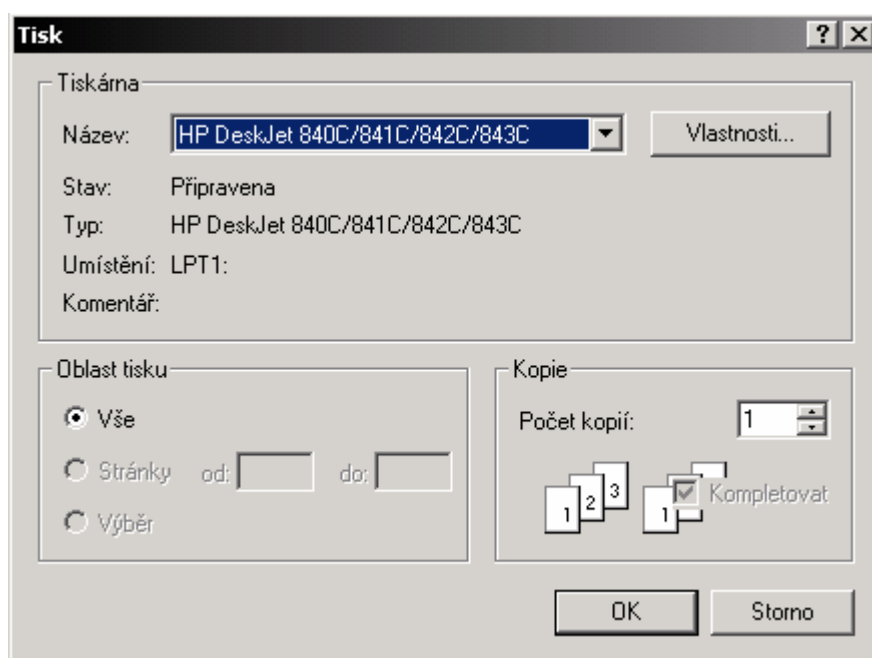
Postup při identifikaci naměřeného průběhu

1. Otevřete soubor s daty, které byly uloženy tímto programem v sekci *Dynamické měření*.
2. Pokud je potřeba, odstraňte z naměřených hodnot ta data, která jsou viditelně nepřesná a nehodí se pro další zpracování. Odstranění se provede tak, že vyberete řádek v tabulce naměřených hodnot, který chcete odstranit a stisknete tlačítko

Odstranit. Data se odstraní z tabulky a zároveň z grafu. Takto upravená data lze poté uložit tlačítkem *Uložit data*.

3. Vyplňte vlastnosti kapaliny v bloku *Vlastnosti materiálu*, přednastavené hodnoty viskozity a hustoty jsou vlastnosti vody při normálním tlaku a teplotě 20°C. Hodnota *Šum* vyjadřuje dynamické podmínky měření.
4. Nastavte typ aproximace, který chcete využít pro parametrizaci, a stiskněte tlačítko *Analyzuj*. Tím se provede identifikace. Výsledné hodnoty parametrů R a m_{MAX} se zobrazí v okně programu a proložený průběh vzlínavosti se zobrazí v grafu.

A.5. Tisk a nastavení tiskárny



Obrázek A.6. Dialog nastavení tisku.

Program *Kern* umožňuje tisknout oblasti grafů na tiskárně. Po stisknutí dialogového tlačítka *Tisk* na kartě *Matematická analýza* se otevře standardní dialog tisku, ve kterém uživatel může vybrat aktivní tiskárnu a nastavovat vlastnosti tisku, jako je barevnost a kvalita tisku, tisk na výšku a na šířku a podobně. Množství nastavitelných vlastností závisí na použité tiskárně. Po potvrzení dialogu tlačítkem *OK* se provede tisk příslušné oblasti grafu.

A.6. Hardwarové nároky

Program je navržen tak, aby uspokojivě a spolehlivě pracoval na počítači s následujícími parametry:

- Procesor minimálně Pentium.
- 16 MB RAM.
- Minimální rozlišení obrazovky je 800x600, doporučuji 1024x768 a výš.
- Sériový port.
- Windows 95 a vyš.

A.7. Elektronický nosič

K projektu je přiloženo CD s následnou adresářovou strukturou:

- \aplikace\bin – zkompilovaná aplikace *KERN*.
- \aplikace\source – zdrojový kód k aplikaci *KERN*.
- \dokument\dp – tato práce ve formátu *PDF*.
- \dokument\prirucka – uživatelská příručka připravená pro tisk.
- \acrobat – instalace *Adobe Acrobat Reader*.

Příloha B. Důležité části kódu

B.1. Připojení sériového kanálu

```
procedure TForm1.PripojPort(Port: PChar; Rychlost: integer;
Parita: integer);
var
  Dcb: TDcb;
  Err: DWORD;
  CStat: TComStat;
  b: Byte;
  r: DWORD;
  s: String;
begin
  if not FCom then //pokud neni pripojen port
  begin
    HCom:=CreateFile(Port,GENERIC_READ or GENERIC_WRITE, 0, nil,
OPEN_EXISTING, 0, 0);
    if HCom<>INVALID_HANDLE_VALUE then //podarilo se otevrit port
    begin
      GetCommState(hCom,dcb);
      with Dcb do //nastaveni DCB struktury
      begin
        case Rychlost of
          0: BaudRate:=150;
          1: BaudRate:=300;
          2: BaudRate:=600;
          3: BaudRate:=1200;
          4: BaudRate:=2400;
          5: BaudRate:=4800;
          6: BaudRate:=9600;
          7: BaudRate:=19200;
        end;
        ByteSize:=7;
        StopBits:=0;
        case Parita of
          0: Parity:=MARKPARITY;
          1: Parity:=SPACEPARITY;
          2: Parity:=ODDPARITY;
          3: Parity:=EVENPARITY;
        end;
        fcom:=SetCommState(Hcom,dcb); //pripojeni portu
        SetupComm(HCom,2048,2048);
      end;{with}
    end;
  end;
  if fCom then begin
    ClearCommError(HCom,Err,@CStat); //vyprazdni vstupni frontu
    while CStat.cbInQue>0 do
    begin
      ReadFile(HCom,b,1,r,nil);
      while b<>10 do begin
        s:=s+char(b);
```

```

        ClearCommError(HCom, Err, @CStat);
        ReadFile(HCom, b, 1, r, nil);
    end;
end;
statusbar1.Panels[0].Text:=Port+' : Připojeno';
end else begin
    statusbar1.Panels[0].Text:=Port+' : Není připojen';
end;
end;
end;

```

B.2. Sběr dat ze sériového kanálu

```

procedure TForm1.Timer1Timer(Sender: TObject);
var
    Err:DWORD;
    CStat:TComStat;
    b:byte;
    r:DWORD;
    s:string;
begin
    if FCom then //pokud je připojen port
    begin
        ClearCommError(HCom, Err, @CStat);
        while CStat.cbInQue>0 do //smaze vstupni frontu a precte
posledni zaznam
        begin
            s:='';
            ReadFile(HCom, b, 1, r, nil);
            while b<>10 do begin
                s:=s+char(b);
                ClearCommError(HCom, Err, @CStat);
                ReadFile(HCom, b, 1, r, nil);
            end;
        end;
        if (s<>'')and(pos('L',s)=0) then begin //pokud to není
chybové hlášení
            Mereni.PosledniHodnota:=DekodujHodnotu(s);
            Panel1.Caption:=FloatToStr(Mereni.PosledniHodnota);
            Panel3.Caption:=Panel1.Caption;
            if CheckBox1.Checked then begin //nepretržite ukládání
                Button1Click(self);
            end;
        end;
    end;
end;
end;

```

B.3. Konverze dat do formátu s plovoucí desetinnou čárkou

```

function TForm1.DekodujHodnotu(s: string): real;
var
    //formátuje vstupní data do float formátu
    s2: string;
    i, a: integer;
begin
    s2:=Copy(s, 0, Length(s)-4);
    while Pos(' ', s2)<>0 do begin

```

```

        i:=Pos(' ', s2);
        s2:=Copy(s2,0,i-1)+Copy(s2,i+1,Length(s2));
    end;
    a:=Pos('.',s2);
    if a<>0 then begin
        s2[a]:=' ';
    end;
    result:=StrToFloat(s2);
end;

```

B.4. Událostní kód dialogového tlačítka *Zaznamenat*

```

procedure TForm1.Button1Click(Sender: TObject);
begin //ulozi hodnotu do stringgrid1
    StringGrid1.RowCount:=StringGrid1.RowCount + 1;
    StringGrid1.FixedRows:=1;
    StringGrid1.Visible:=true;
    Button2.Enabled:=true;
    StringGrid1.Cells[1,0]:='Èas od zaèátku [s]';
    StringGrid1.Cells[2,0]:='Hodnota [g]';
    StringGrid1.Cells[3,0]:='Poznámky';
    StringGrid1.Cells[0,StringGrid1.RowCount-
1]:=IntToStr(StringGrid1.RowCount-1);
    StringGrid1.Cells[1,StringGrid1.RowCount-
1]:=DateTimeToStr(Date+Time);
    StringGrid1.Cells[2,StringGrid1.RowCount-1]:=Panel1.Caption;
end;

```

B.5. Kód metody nejmenších čtverců

```

procedure TForm1.regrese(A, mmax: real;var R, sigma, minY: real;
var minX: integer; Aprox: Boolean);
var i: integer;
    citatel, jmenovatel: real;
    maxX: integer;
    iks: real;
begin
    citatel:=0;
    jmenovatel:=0;
    if Aprox then begin //pouzi je odmocninovou aproximaci
        maxX:=1;//zjisteni casu konce identifikace (namerena hodnota
na 40 procentech)
        while Chart2.SeriesList[0].YValue[maxX]<0.4*(mmax-minY)+minY
do begin
            maxX:=maxX+1;
        end;
        for i:=minX-1 to maxX do begin
            citatel:=citatel+(Chart2.SeriesList[0].XValue[i]-
Chart2.SeriesList[0].XValue[minX])
                *sqr(Chart2.SeriesList[0].YValue[i]-minY)/A;
            jmenovatel:=jmenovatel+sqr(Chart2.SeriesList[0].XValue[i]-
Chart2.SeriesList[0].XValue[minX]);
        end;
        R:=sqrt(abs(citatel/jmenovatel/(mmax-minY)));
    end;

```

```

sigma:=0;
for i:=minX-1 to maxX do begin
    sigma:=sigma+sqr (Chart2.SeriesList[0].YValue[i]
        -sqr (A*R*R* (mmax-minY) * (Chart2.SeriesList[0].XValue[i]-
Chart2.SeriesList[0].XValue[minX-1])));
end;
end else begin

maxX:=Chart2.Series[0].Count-1;
for i:=minX-1 to maxX do begin
    iks:=-Chart2.SeriesList[0].YValue[i]+minY+
        (mmax-minY)*ln(abs((mmax-minY)/(mmax-
Chart2.SeriesList[0].YValue[i])));
    citatel:=citatel+(Chart2.SeriesList[0].XValue[i]-
Chart2.SeriesList[0].XValue[minX])
        *iks/A;
    jmenovatel:=jmenovatel+sqr (Chart2.SeriesList[0].XValue[i]-
Chart2.SeriesList[0].XValue[minX]);
end;
R:=sqr (citatel/jmenovatel);
sigma:=0;
for i:=minX-1 to maxX do begin
    sigma:=sigma+sqr (Chart2.SeriesList[0].YValue[i]
        - (minX+ (mmax-minY)*sqr (1-exp(-
        (Chart2.SeriesList[0].XValue[i]-
Chart2.SeriesList[0].XValue[minX-1])
        *A*sqr (R)/(mmax-minY)/0.65)))));
end;
end;
end;
end;

```

B.6. Kód metody dělení intervalu

```

while (mmax2-mmax1)>0.01 do begin //metoda tretinoveho intervalu
    mmax13:=mmax1+(mmax2-mmax1)/3;
    mmax23:=mmax1+(mmax2-mmax1)*2/3;
    regrese(A, mmax13, R, sigma1, minY, minX,
        RadioGroup1.ItemIndex=0);
    regrese(A, mmax23, R, sigma2, minY, minX,
        RadioGroup1.ItemIndex=0);
    if sigma1>sigma2 then begin
        mmax1:=mmax13;
    end else begin
        mmax2:=mmax23;
    end;
end;
end;

```

Příloha C. Elektronický nosič

K projektu je přiloženo CD s následnou adresářovou strukturou:

- \aplikace\bin – zkompilovaná aplikace *KERN*.
- \aplikace\source – zdrojový kód k aplikaci *KERN*.
- \dokument\dp – tato práce ve formátu *PDF*.
- \dokument\prirucka – uživatelská příručka připravená pro tisk.
- \acrobat – instalace *Adobe Acrobat Reader*.

Použitá literatura

- [1] Petr Kocourek a kolektiv: *Číslicové měřicí systémy*, ČVUT 1994.
- [2] Robert Clemenzi: *Serial I/O (RS232) Using Delphi (RS232)*,
http://www.cpcug.org/user/clemenzi/technical/Languages/Delphi/Delphi_SerialIO.htm.
- [3] Kern & Sohn GmbH: manuál k váhám *KERN 770/GS/GJ*, www.kern-sohn.de 04/2000.
- [4] Karel Rektorys a spolupracovníci: *Přehled užití matematiky*, SNTL 1963.
- [5] Microsoft: *MSDN*, <http://msdn.microsoft.com/>.
- [6] Inprise: *Windows SDK*, dodáváno se systémem *Inprise Delphi*.
- [7] Inprise: *Nápověda k Inprise Delphi 5*, dodáváno se systémem *Inprise Delphi*.
- [8] Ing. Jakub Wiener Ph.D: *Dynamika procesu vztlínání kapaliny* (průběžná zpráva), listopad 2002.